Gaussian Processes for Uncertainty Quantification
Part I

Javier González
Amazon Cambridge, UK

MLSS, Buenos Aires, Argentina

19th Jun 2018

The world is an uncertain place

What do we understand by uncertainty?

Result of a live game

Aleatoric uncertainty

Result of a recorded game

Epistemic uncertainty

Uncertainty is lack of knowledge

How do we quantify uncertainty?

Probability is the universal language of (any) uncertainty

Two simple rules:

- *Sum rule*: $p(x) = \sum_y p(x, y)$
- *Product rule*: $p(x, y) = p(y|x)p(x)$

$$p(\theta|Data) = \frac{p(Data|\theta)p(\theta)}{p(Data)}$$

for θ some unknown latent quantity and *Data* (observables).

How do we quantify uncertainty?

Probability is the universal language of (any) uncertainty

Two simple rules:

- *Sum rule*: $p(x) = \sum_y p(x, y)$
- *Product rule*: $p(x, y) = p(y|x)p(x)$

$$p(\theta|Data)p(Data) = p(Data|\theta)p(\theta)$$

for θ some unknown latent quantity and *Data* (observables).

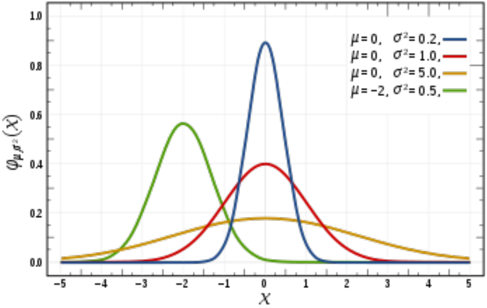
1. **Probability helps to model any source of uncertainty**

Outline of the lecture

- Part I: **Introduction to Gaussian processes**
  - Basic description of Gaussian processes.
  - Gaussian processes with non Gaussian likelihoods.
  - Functional point of view on Gaussian processes and connections.
  - Deep Gaussian processes.
- Part II: *Decision making under uncertainty*
  - General framework for decision making.
  - Bayesian optimization.
  - Bayesian quadrature.
  - Experimental design.

The Gaussian distribution

$$\varphi_{\mu,\sigma^2}(y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y-\mu)^2}{2\sigma^2}}$$



Why is it so famous? Why does it have this shape?

Why the Gaussian?

- **Central limit theorem**: Sums of independent random variables with finite mean and variance are asymptotically Gaussian.

- **Maximum entropy**: The Gaussian has maximum entropy relative to all probability distributions covering the entire real line with finite mean and variance.

- **The Laplace approximation:** Uses the Taylor expansion to approximate an arbitrary distribution at the mode.

- **Linear algebra:** Computations with the Gaussian are linear algebra operations.

Why the Gaussian?

- **Central limit theorem**: Sums of independent random variables with finite mean and variance are asymptotically Gaussian.

- **Maximum entropy**: The Gaussian has maximum entropy relative to all probability distributions covering the entire real line with finite mean and variance.

- **The Laplace approximation:** Uses the Taylor expansion to approximate an arbitrary distribution at the mode.

- **Linear algebra:** Computations with the Gaussian are linear algebra operations.

Why the Gaussian?

- **Central limit theorem**: Sums of independent random variables with finite mean and variance are asymptotically Gaussian.

- **Maximum entropy**: The Gaussian has maximum entropy relative to all probability distributions covering the entire real line with finite mean and variance.

- **The Laplace approximation:** Uses the Taylor expansion to approximate an arbitrary distribution at the mode.

- **Linear algebra:** Computations with the Gaussian are linear algebra operations.

Why the Gaussian?

- **Central limit theorem**: Sums of independent random variables with finite mean and variance are asymptotically Gaussian.

- **Maximum entropy**: The Gaussian has maximum entropy relative to all probability distributions covering the entire real line with finite mean and variance.

- **The Laplace approximation:** Uses the Taylor expansion to approximate an arbitrary distribution at the mode.

- **Linear algebra:** Computations with the Gaussian are linear algebra operations.

1. **Probability helps to model any source of uncertainty**
2. **The Gaussian appears in many operations in science**

Two important Gaussian properties

Sum of Gaussians

- Sum of independent Gaussian variables is also Gaussian.

$$y_i \sim \mathcal{N}\left(\mu_i, \sigma_i^2\right)$$

And the sum is distributed as

$$\sum_{i=1}^{n} y_i \sim \mathcal{N}\left(\sum_{i=1}^{n} \mu_i, \sum_{i=1}^{n} \sigma_i^2\right)$$

(*Aside*: The central limit theorem also holds.)

Two Important Gaussian properties

Scaling a Gaussian

- Scaling a Gaussian leads to a Gaussian.

$$y \sim \mathcal{N}\left(\mu, \sigma^2\right)$$

And the scaled density is distributed as

$$wy \sim \mathcal{N}\left(w\mu, w^2\sigma^2\right)$$

Two dimensional Gaussian

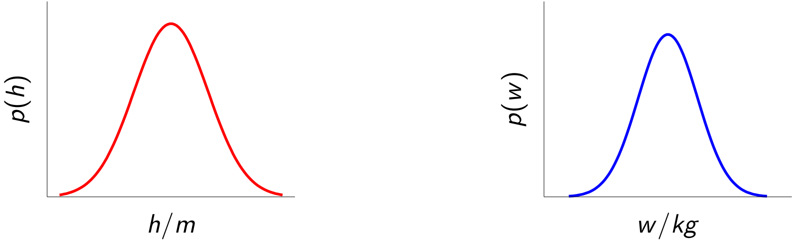
- Consider height, h/m and weight, w/kg.
- Could sample height from a distribution:

$$p(h) \sim \mathcal{N}(1.7, 0.0225)$$

- And similarly weight:

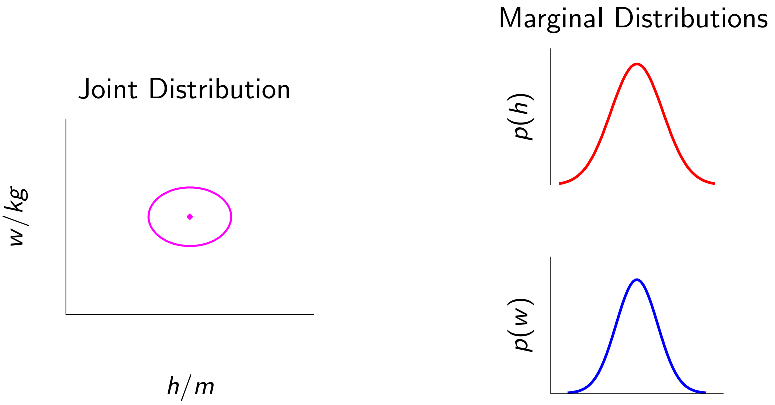
$$p(w) \sim \mathcal{N}(75, 36)$$

Height and weight models



Gaussian distributions for height and weight.

Sampling two dimensional variables

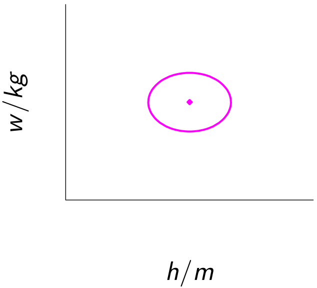


Joint Distribution

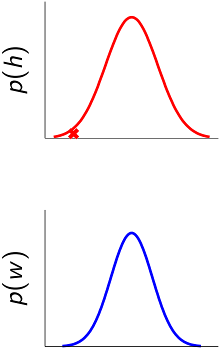
Marginal Distributions

Samples of height and weight

Sampling two dimensional variables



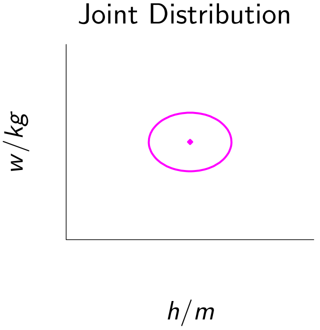
Joint Distribution

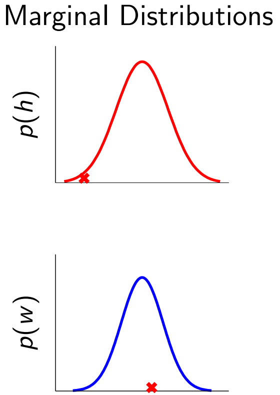


Marginal Distributions

Samples of height and weight

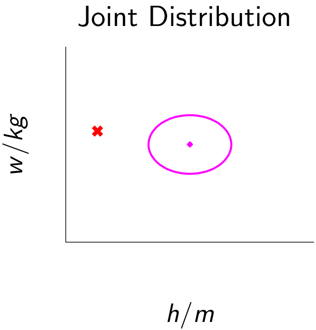
Sampling two dimensional variables

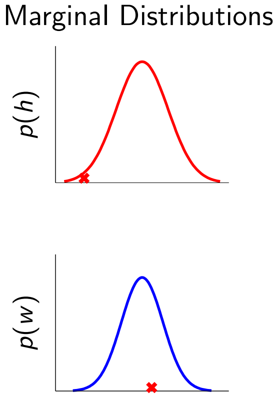




Samples of height and weight

Sampling two dimensional variables

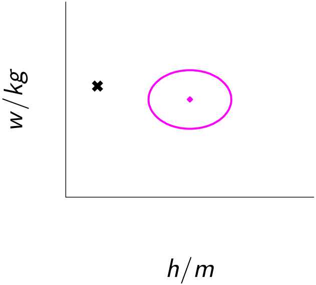




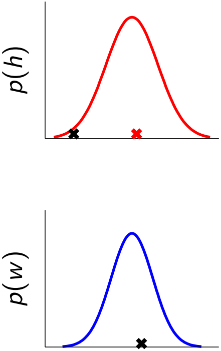
Samples of height and weight

Sampling two dimensional variables

Joint Distribution

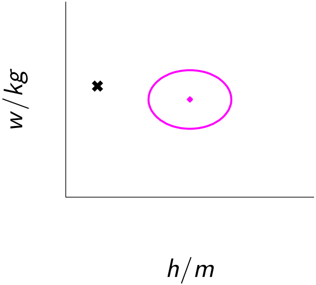


Marginal Distributions

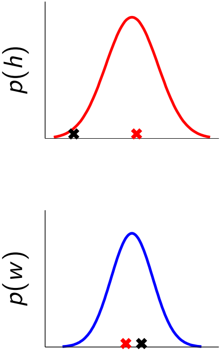


Samples of height and weight

Sampling two dimensional variables



Joint Distribution

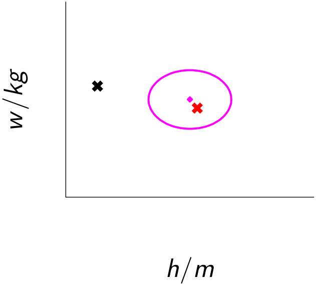


Marginal Distributions

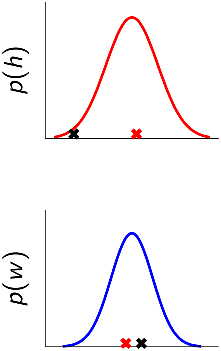
Samples of height and weight

Sampling two dimensional variables

Joint Distribution

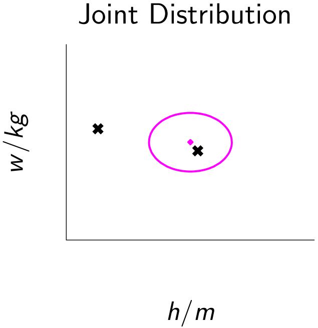


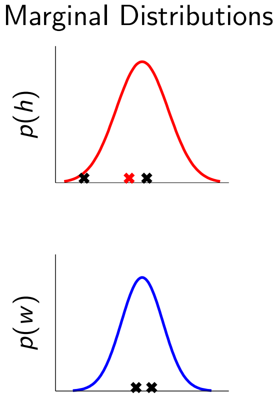
Marginal Distributions



Samples of height and weight

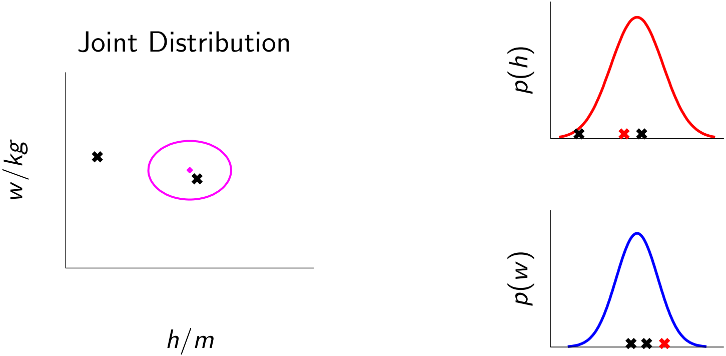
Sampling two dimensional variables





Samples of height and weight

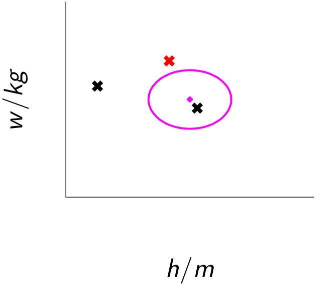
Sampling two dimensional variables

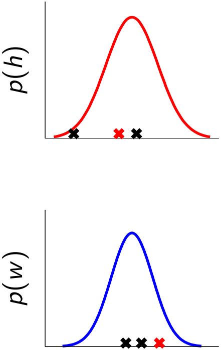

Joint Distribution

Marginal Distributions

Samples of height and weight

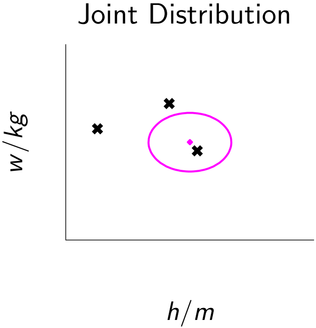
Sampling two dimensional variables

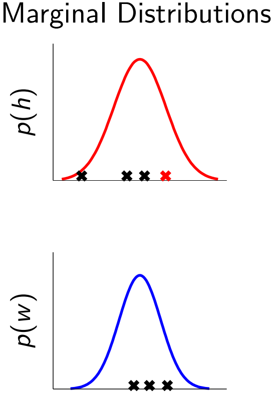

Joint Distribution


Marginal Distributions

Samples of height and weight

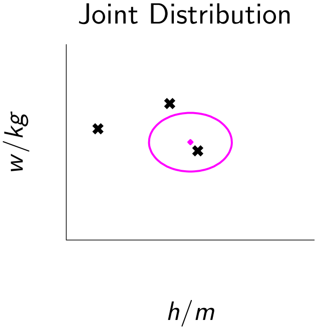
Sampling two dimensional variables

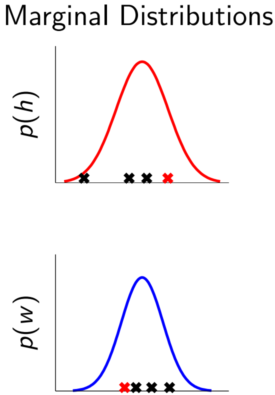




Samples of height and weight

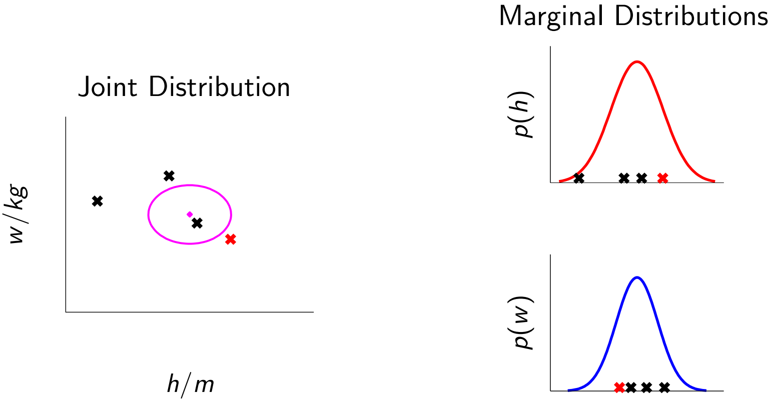
Sampling two dimensional variables





Samples of height and weight

Sampling two dimensional variables



Joint Distribution

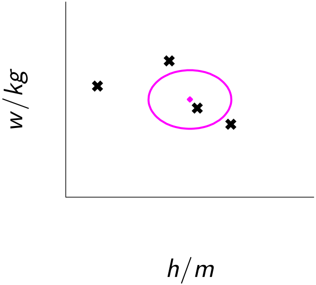
Marginal Distributions

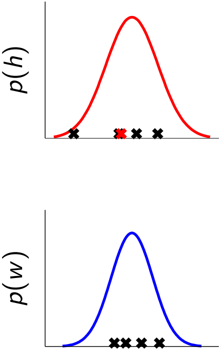
Samples of height and weight

Sampling two dimensional variables

Marginal Distributions

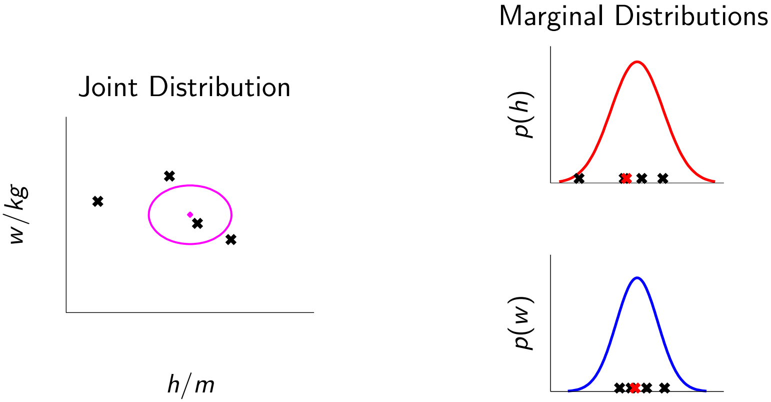
Joint Distribution





Samples of height and weight

Sampling two dimensional variables

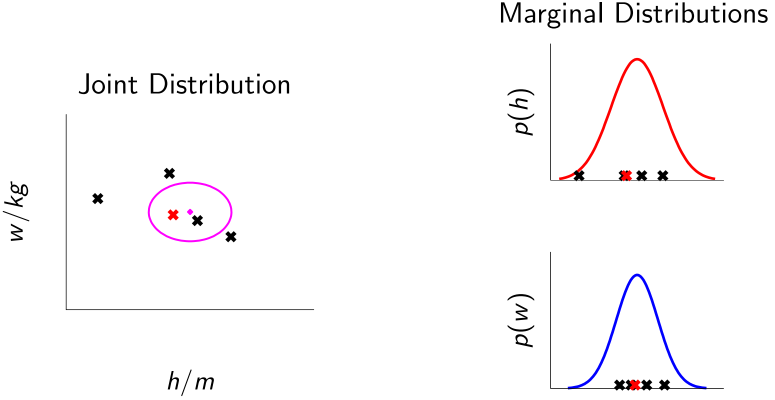


Joint Distribution

Marginal Distributions

Samples of height and weight

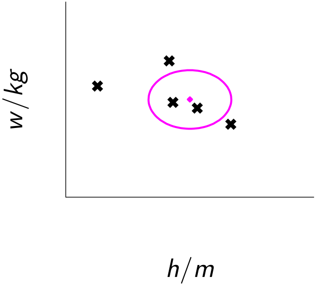
Sampling two dimensional variables



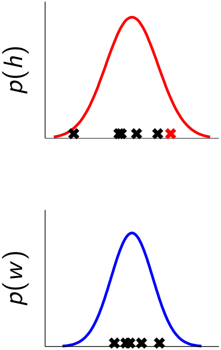
Samples of height and weight

Sampling two dimensional variables

Joint Distribution

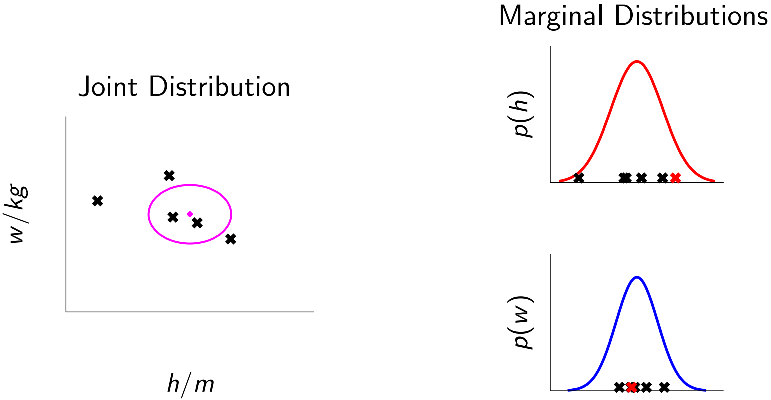


Marginal Distributions



Samples of height and weight

Sampling two dimensional variables

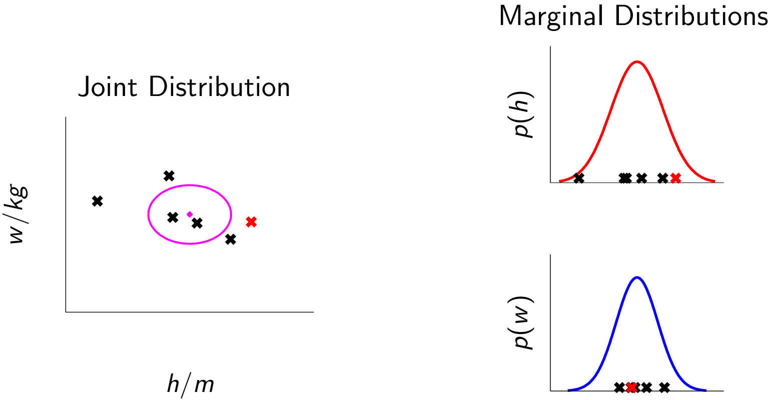


Joint Distribution

Marginal Distributions

Samples of height and weight

Sampling two dimensional variables

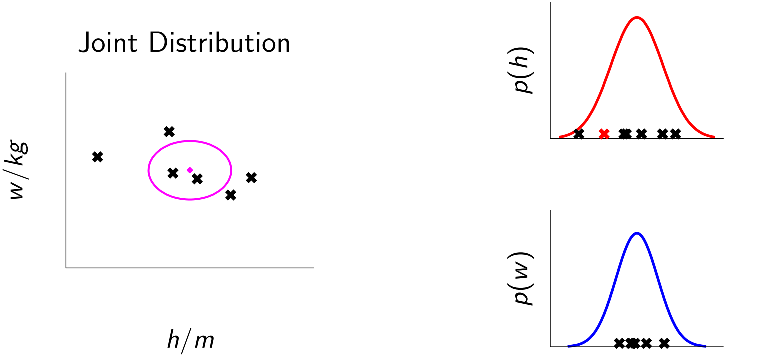


Marginal Distributions

Joint Distribution

Samples of height and weight

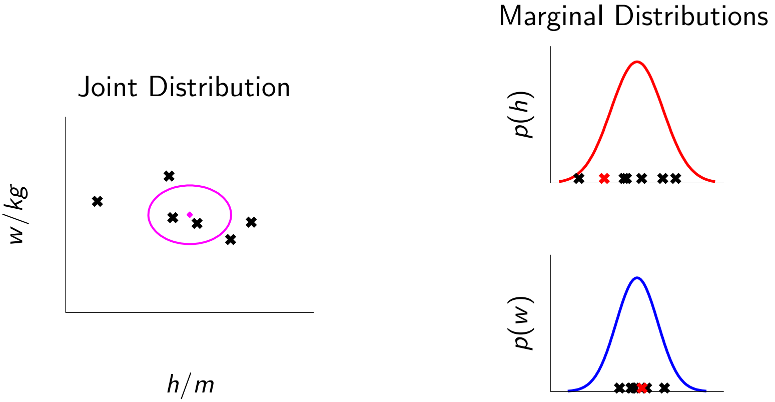
Sampling two dimensional variables


Joint Distribution

Marginal Distributions

Samples of height and weight

Sampling two dimensional variables

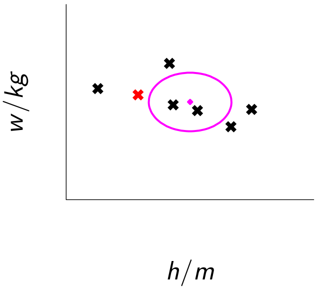


Joint Distribution

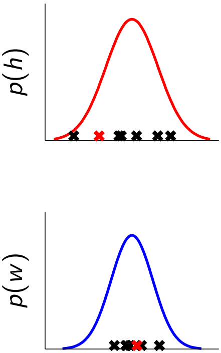
Marginal Distributions

Samples of height and weight

Sampling two dimensional variables



Joint Distribution

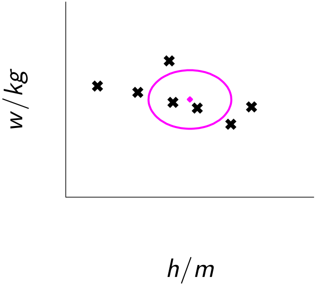


Marginal Distributions

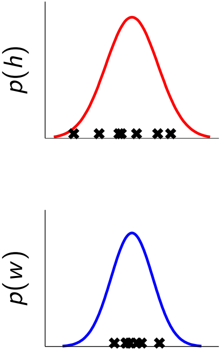
Samples of height and weight

Sampling two dimensional variables

Joint Distribution



Marginal Distributions



Samples of height and weight

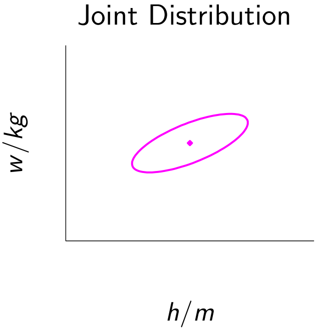
Independence assumption

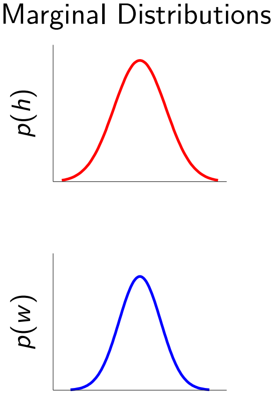
- This assumes height and weight are independent.

$$p(h, w) = p(h)p(w)$$

- In reality they are dependent (body mass index) $= \frac{w}{h^2}$.

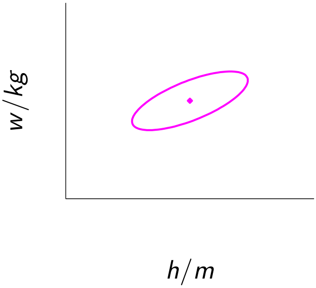
Sampling two dimensional variables



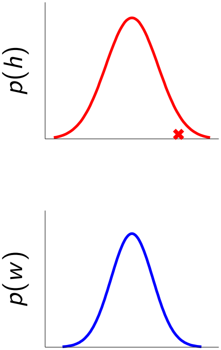


Sampling two dimensional variables

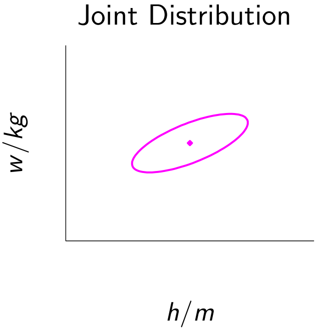
Joint Distribution

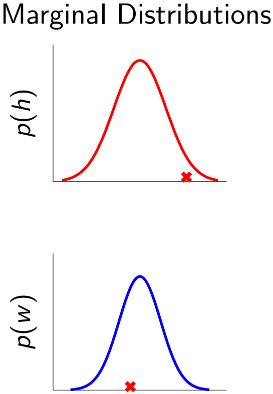


Marginal Distributions



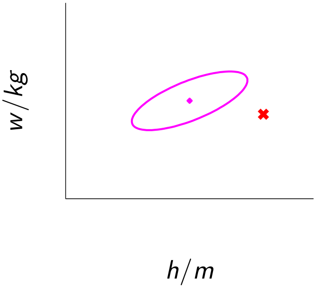
Sampling two dimensional variables


Joint Distribution

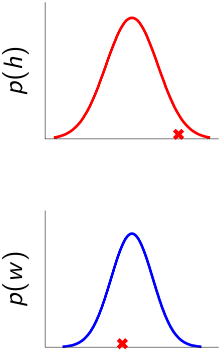

Marginal Distributions

Sampling two dimensional variables

Joint Distribution

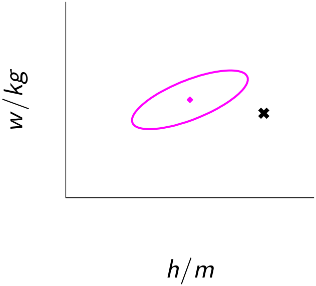


Marginal Distributions

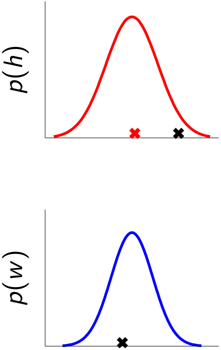


Sampling two dimensional variables

Joint Distribution

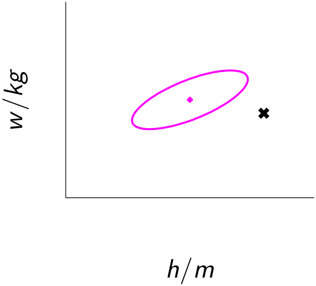


Marginal Distributions

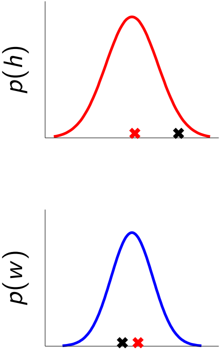


Sampling two dimensional variables

Joint Distribution

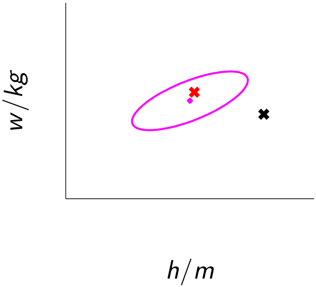


Marginal Distributions

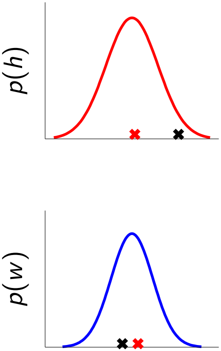


Sampling two dimensional variables

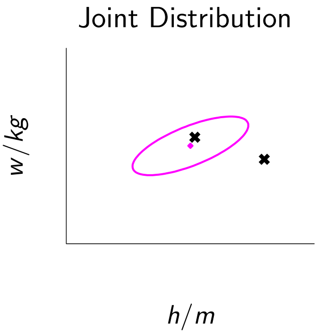
Joint Distribution

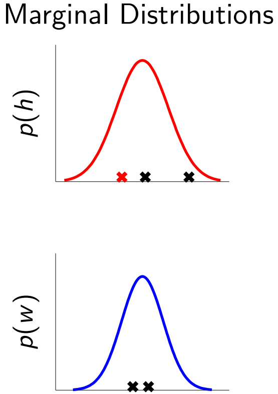


Marginal Distributions

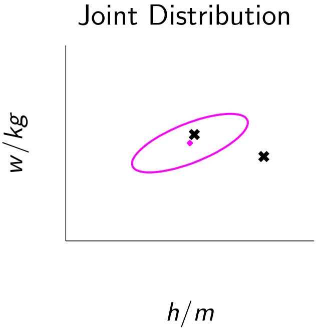


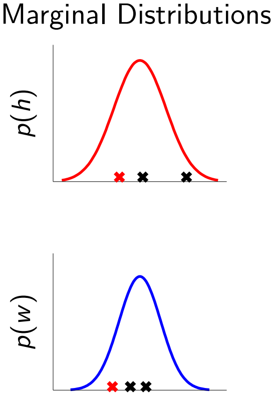
Sampling two dimensional variables



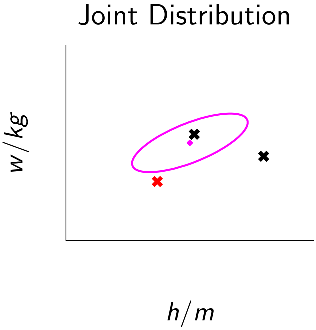


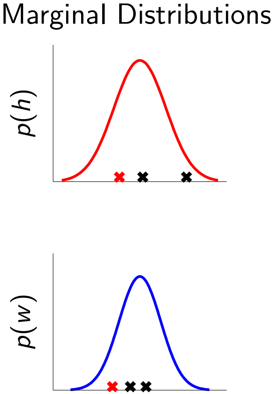
Sampling two dimensional variables





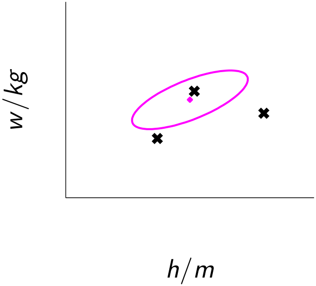
Sampling two dimensional variables


Joint Distribution

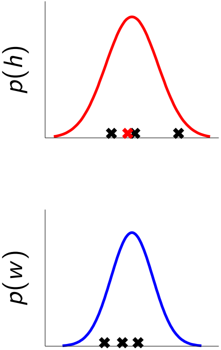

Marginal Distributions

Sampling two dimensional variables

Joint Distribution

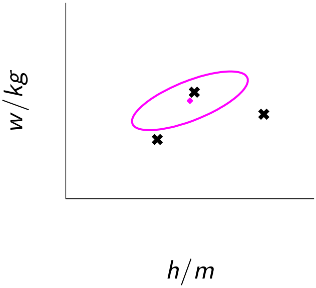


Marginal Distributions

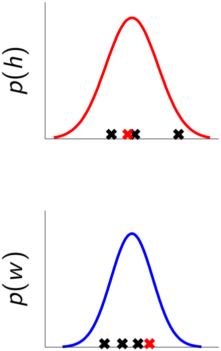


Sampling two dimensional variables

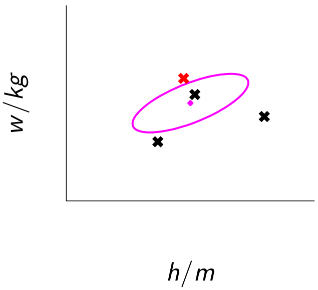
Joint Distribution



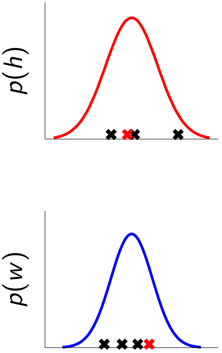
Marginal Distributions



Sampling two dimensional variables

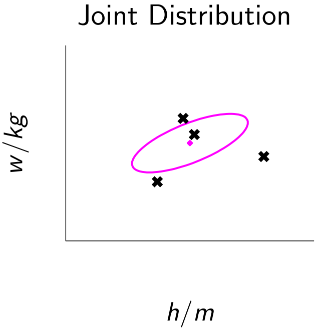


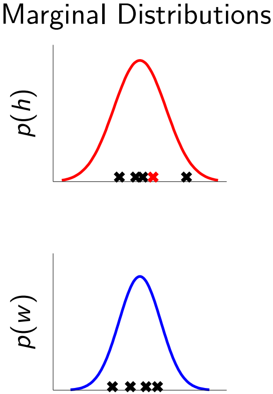
Joint Distribution



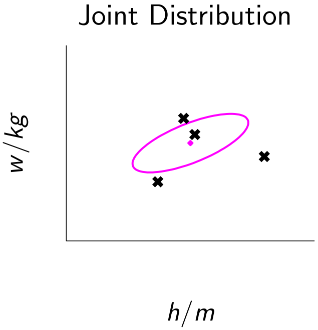
Marginal Distributions

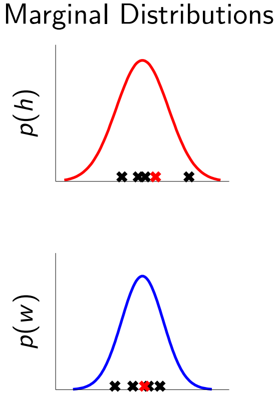
Sampling two dimensional variables





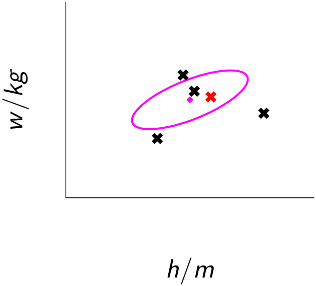
Sampling two dimensional variables



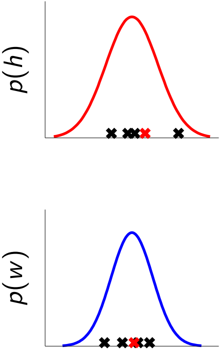


Sampling two dimensional variables

Joint Distribution

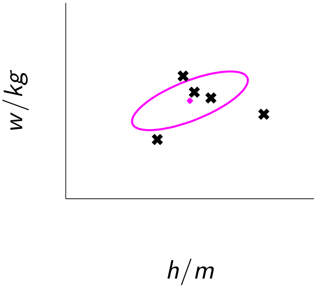


Marginal Distributions

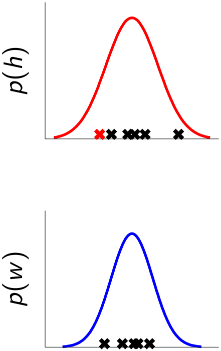


Sampling two dimensional variables

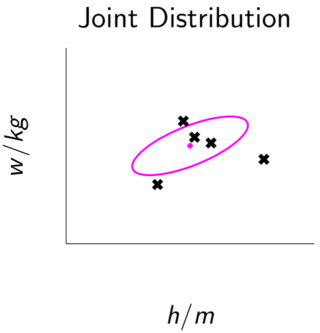
Joint Distribution

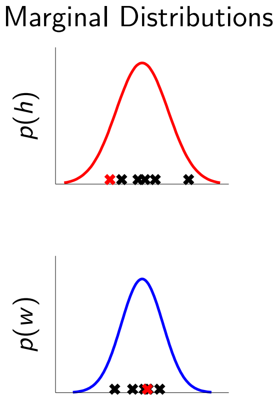


Marginal Distributions

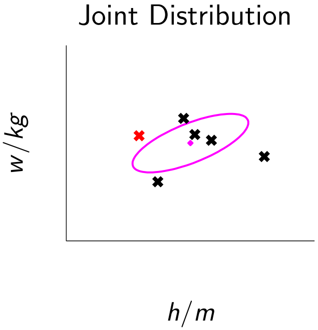


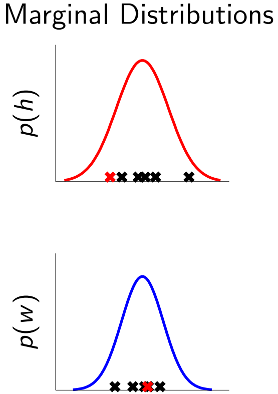
Sampling two dimensional variables


Joint Distribution


Marginal Distributions

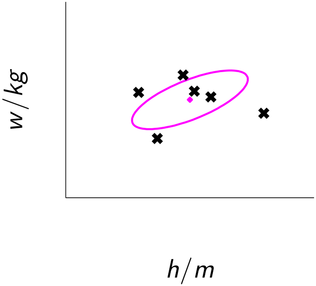
Sampling two dimensional variables



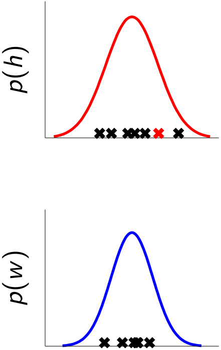


Sampling two dimensional variables

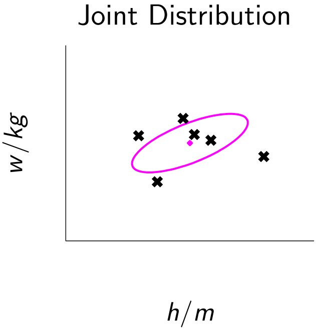
Joint Distribution

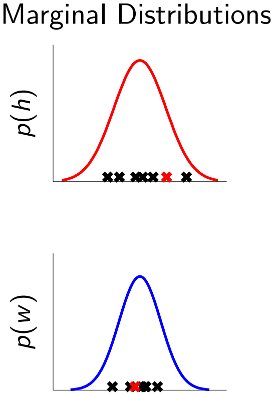


Marginal Distributions



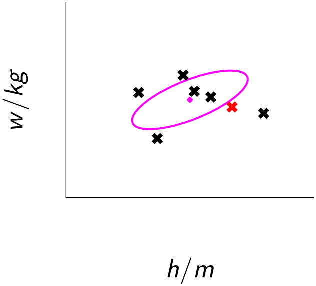
Sampling two dimensional variables



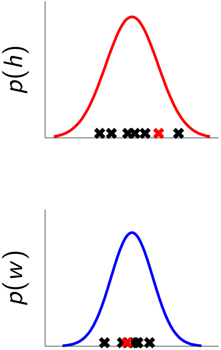


Sampling two dimensional variables

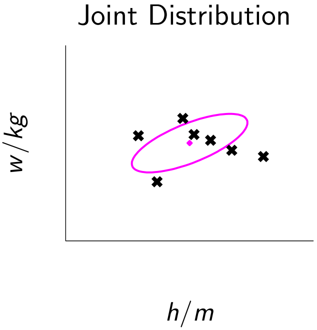
Joint Distribution

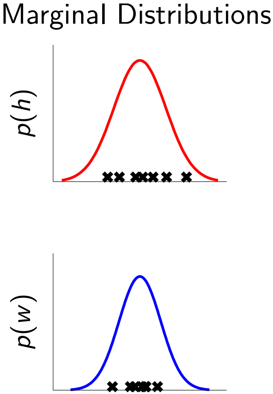


Marginal Distributions



Sampling two dimensional variables





Independent Gaussians

$$p(w, h) = p(w)p(h)$$

$$p(w, h) = \frac{1}{\sqrt{2\pi\sigma_1^2}\sqrt{2\pi\sigma_2^2}} \exp\left(-\frac{1}{2}\left(\frac{(w-\mu_1)^2}{\sigma_1^2} + \frac{(h-\mu_2)^2}{\sigma_2^2}\right)\right)$$

$$p(w, h) = \frac{1}{2\pi\sigma_1\sigma_2} \exp\left(-\frac{1}{2}\left(\begin{bmatrix} w \\ h \end{bmatrix} - \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}\right)^\top \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}^{-1} \left(\begin{bmatrix} w \\ h \end{bmatrix} - \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}\right)\right)$$

$$p(\mathbf{y}) = \frac{1}{2\pi\left|\mathbf{D}\right|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{y}-\boldsymbol{\mu})^\top \mathbf{D}^{-1}(\mathbf{y}-\boldsymbol{\mu})\right)$$

Independent Gaussians

$$p(w, h) = p(w)p(h)$$

$$p(w, h) = \frac{1}{\sqrt{2\pi\sigma_1^2}\sqrt{2\pi\sigma_2^2}} \exp\left(-\frac{1}{2}\left(\frac{(w-\mu_1)^2}{\sigma_1^2} + \frac{(h-\mu_2)^2}{\sigma_2^2}\right)\right)$$

$$p(w, h) = \frac{1}{2\pi\sigma_1\sigma_2} \exp\left(-\frac{1}{2}\left(\begin{bmatrix} w \\ h \end{bmatrix} - \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}\right)^\top \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}^{-1} \left(\begin{bmatrix} w \\ h \end{bmatrix} - \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}\right)\right)$$

$$p(\mathbf{y}) = \frac{1}{2\pi\left|\mathbf{D}\right|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{y}-\boldsymbol{\mu})^\top \mathbf{D}^{-1}(\mathbf{y}-\boldsymbol{\mu})\right)$$

Independent Gaussians

$$p(w, h) = p(w)p(h)$$

$$p(w, h) = \frac{1}{\sqrt{2\pi\sigma_1^2}\sqrt{2\pi\sigma_2^2}} \exp\left(-\frac{1}{2}\left(\frac{(w-\mu_1)^2}{\sigma_1^2} + \frac{(h-\mu_2)^2}{\sigma_2^2}\right)\right)$$

$$p(w, h) = \frac{1}{2\pi\sigma_1\sigma_2} \exp\left(-\frac{1}{2}\left(\begin{bmatrix} w \\ h \end{bmatrix} - \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}\right)^\top \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}^{-1} \left(\begin{bmatrix} w \\ h \end{bmatrix} - \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}\right)\right)$$

$$p(\mathbf{y}) = \frac{1}{2\pi\left|\mathbf{D}\right|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{y}-\boldsymbol{\mu})^\top \mathbf{D}^{-1}(\mathbf{y}-\boldsymbol{\mu})\right)$$

Independent Gaussians

$$p(w, h) = p(w)p(h)$$

$$p(w, h) = \frac{1}{\sqrt{2\pi\sigma_1^2}\sqrt{2\pi\sigma_2^2}} \exp\left(-\frac{1}{2}\left(\frac{(w-\mu_1)^2}{\sigma_1^2} + \frac{(h-\mu_2)^2}{\sigma_2^2}\right)\right)$$

$$p(w, h) = \frac{1}{2\pi\sigma_1\sigma_2} \exp\left(-\frac{1}{2}\left(\begin{bmatrix} w \\ h \end{bmatrix} - \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}\right)^\top \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}^{-1} \left(\begin{bmatrix} w \\ h \end{bmatrix} - \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}\right)\right)$$

$$p(\mathbf{y}) = \frac{1}{2\pi\left|\mathbf{D}\right|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{y}-\boldsymbol{\mu})^\top \mathbf{D}^{-1}(\mathbf{y}-\boldsymbol{\mu})\right)$$

Correlated Gaussian

Form a correlated Gaussian from original by rotating the data space using matrix $\mathbf{R}$.

$$p(\mathbf{y}) = \frac{1}{2\pi \left|\mathbf{D}\right|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{y}-\boldsymbol{\mu})^\top \mathbf{D}^{-1}(\mathbf{y}-\boldsymbol{\mu})\right)$$

Correlated Gaussian

Form a correlated Gaussian from original by rotating the data space using matrix $\mathbf{R}$.

$$p(\mathbf{y}) = \frac{1}{2\pi \left|\mathbf{R}\mathbf{D}\mathbf{R}^T\right|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{R}^\top \mathbf{y} - \mathbf{R}^\top \boldsymbol{\mu})^\top \mathbf{D}^{-1}(\mathbf{R}^\top \mathbf{y} - \mathbf{R}^\top \boldsymbol{\mu})\right)$$

Correlated Gaussian

Form a correlated Gaussian from original by rotating the data space using matrix **R**.

$$p(\mathbf{y}) = \frac{1}{2\pi \left|\mathbf{R}\mathbf{D}\mathbf{R}^{T}\right|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{y}-\boldsymbol{\mu})^{\top}\mathbf{R}\mathbf{D}^{-1}\mathbf{R}^{\top}(\mathbf{y}-\boldsymbol{\mu})\right)$$

this gives a covariance matrix:

$$\mathbf{C} = \mathbf{R}\mathbf{D}\mathbf{R}^{\top}$$

$$\mathbf{C}^{-1} = \mathbf{R}\mathbf{D}^{-1}\mathbf{R}^{\top}$$

Correlated Gaussian

Form a correlated Gaussian from original by rotating the data space using matrix $\mathbf{R}$.

$$p(\mathbf{y}) = \frac{1}{2\pi \left|\mathbf{C}\right|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{y}-\boldsymbol{\mu})^{\top}\mathbf{C}^{-1}(\mathbf{y}-\boldsymbol{\mu})\right)$$

this gives a covariance matrix:

$$\mathbf{C} = \mathbf{R}\mathbf{D}\mathbf{R}^{\top}$$

$$\mathbf{C}^{-1} = \mathbf{R}\mathbf{D}^{-1}\mathbf{R}^{\top}$$

Recall univariate Gaussian properties

1. Sum of Gaussian variables is also Gaussian.

$$y_i \sim \mathcal{N}\left(\mu_i, \sigma_i^2\right)$$

$$\sum_{i=1}^{n} y_i \sim \mathcal{N}\left(\sum_{i=1}^{n} \mu_i, \sum_{i=1}^{n} \sigma_i^2\right)$$

2. Scaling a Gaussian leads to a Gaussian.

$$y \sim \mathcal{N}\left(\mu, \sigma^2\right)$$

$$wy \sim \mathcal{N}\left(w\mu, w^2\sigma^2\right)$$

Multivariate consequence

- If

$$\mathbf{x} \sim \mathcal{N}\left(\boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$$

- And

$$\mathbf{y} = \mathbf{W}\mathbf{x}$$

- Then

$$\mathbf{y} \sim \mathcal{N}\left(\mathbf{W}\mu, \mathbf{W}\Sigma\mathbf{W}^{\top}\right)$$

Multivariate consequence

- If

$$\mathbf{x} \sim \mathcal{N}\left(\boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$$

- And

$$\mathbf{y} = \mathbf{W}\mathbf{x}$$

- Then

$$\mathbf{y} \sim \mathcal{N}\left(\mathbf{W}\mu, \mathbf{W}\Sigma\mathbf{W}^\top\right)$$

Multivariate consequence

- If

$$\mathbf{x} \sim \mathcal{N}\left(\boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$$

- And

$$\mathbf{y} = \mathbf{W}\mathbf{x}$$

- Then

$$\mathbf{y} \sim \mathcal{N}\left(\mathbf{W}\boldsymbol{\mu}, \mathbf{W}\boldsymbol{\Sigma}\mathbf{W}^{\top}\right)$$

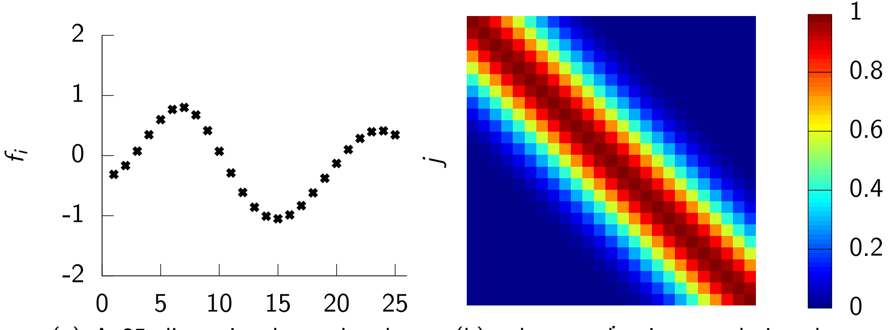
Sampling a multivariate Gaussian

$$\mathbf{f} \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{\Sigma}\right)$$

Multivariate Gaussians

- We will consider a Gaussian with a particular structure of covariance matrix.
- Generate a single sample from this 25 dimensional Gaussian distribution, $\mathbf{f} = [f_1, f_2 \ldots f_{25}]$.
- We will plot these points against their index.

Gaussian distribution sample



(a) A 25 dimensional correlated random variable (values ploted against index)

(b) colormap showing correlations between dimensions.

Figure: A sample from a 25 dimensional Gaussian distribution.

1. **Probability helps to model any source uncertainty**

2. **The Gaussian appears in many operations in science**

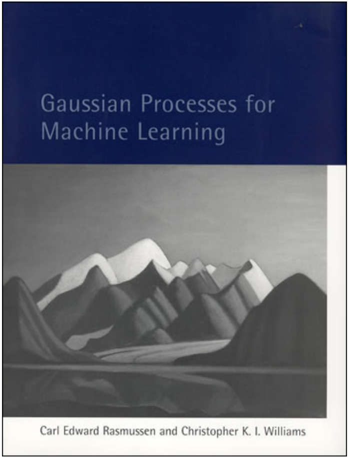
3. **Correlations are key in the samples from the Gaussian**

Gaussian processes

A Gaussian process (GP) is an infinite-dimensional probability density, such that each linear finite-dimensional restriction is multivariate Gaussian.

- Generalization of the multivariate Gaussian to functional spaces.
- Finite collections of variables of Gaussian processes are Gaussian.

Book: Gaussian processes for machine learning


Gaussian Processes for Machine Learning
Carl Edward Rasmussen and Christopher K. I. Williams

The Gaussian processes book: Rasmussen and Williams, 2006

Gaussian process history

- **Geostatistics**: Kriging 1970s, normally used in spatio-temporal modelling.
- **Spatial statistics**: Cressie [1993] for overview.
- **Time series:** Wiener, Kolmogorov 1940s.
- **Uncertainty quantification/computer experiments:** OHagan [1978], Sacks et al. [1989].
- **Signal processing**: Kalman filter is a particular representation of a Gaussian process. See work of Simo Sarkka.

Gaussian processes for regression

- Set of *inputs* $\mathbf{X} = \{x_1, x_2, ..., x_N\}$ corresponding set of random function variables $\mathbf{f} = \{f_1, f_2, \ldots, f_N\}$.

- GP: Any set of function variables $\{f_n\}_{i=1}^{N}$ has joint (zero mean) Gaussian distribution

$$p(\mathbf{f}|\mathbf{X}) \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{K}\right).$$

- In principle, the density of the inputs is not modelled.

Covariance functions

Where does the covariance (kernel) matrix $\mathbf{K}$ comes from?

- The covariance function K represents how we believe the $f(\mathbf{x}_i)$ and $f(\mathbf{x}_j)$ are correlated given $\mathbf{x}_i$ and $\mathbf{x}_j$.
- The covariance matrix $\mathbf{K}$ has elements $\mathbf{K}_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$.
- $\mathbf{K}$ must be positive semi-definite, $\mathbf{a}^T\mathbf{K}\mathbf{a} \geq 0$

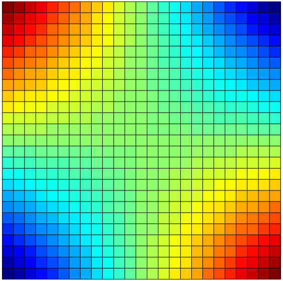
Covariance functions

Linear covariance function

$$k\left(\mathbf{x}, \mathbf{x}'\right) = \alpha \mathbf{x}^\top \mathbf{x}'$$

- Bayesian linear regression.

$$\alpha = 1$$



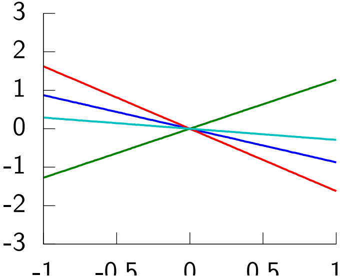
Covariance functions

Linear covariance function

$$k\left(\mathbf{x}, \mathbf{x}'\right) = \alpha \mathbf{x}^{\top} \mathbf{x}'$$

- Bayesian linear regression.

$$\alpha = 1$$



Covariance functions

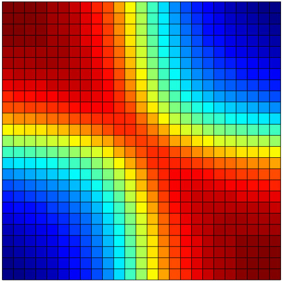
MLP covariance function

$$k\left(\mathbf{x}, \mathbf{x}'\right) = \alpha \mathrm{asin}\left(\frac{w\mathbf{x}^{\top}\mathbf{x}' + b}{\sqrt{w\mathbf{x}^{\top}\mathbf{x} + b + 1}\sqrt{w\mathbf{x}'^{\top}\mathbf{x}' + b + 1}}\right)$$

- Based on infinite neural network model.

$$w = 40$$

$$b = 4$$



Covariance functions

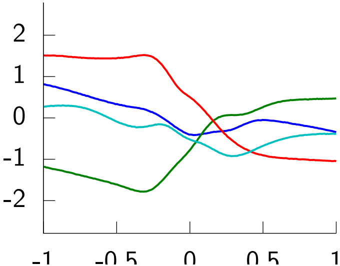
MLP covariance function

$$k\left(\mathbf{x},\mathbf{x}'\right) = \alpha \text{asin}\left(\frac{w\mathbf{x}^\top\mathbf{x}' + b}{\sqrt{w\mathbf{x}^\top\mathbf{x} + b + 1}\sqrt{w\mathbf{x}'^\top\mathbf{x}' + b + 1}}\right)$$

- Based on infinite neural network model.

$$w = 40$$

$$b = 4$$



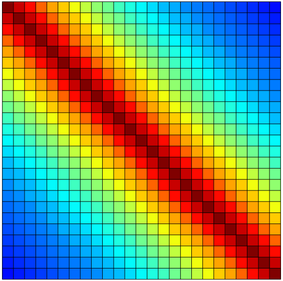
Covariance functions

Where did this covariance matrix come from?

Ornstein-Uhlenbeck (stationary Gauss-Markov) covariance function

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

- One dimension: arises from a stochastic differential equation (Brownian motion in a parabolic tube).
- Higher dimensions: Fourier filter of the form $\frac{1}{\pi(1+x^2)}$.



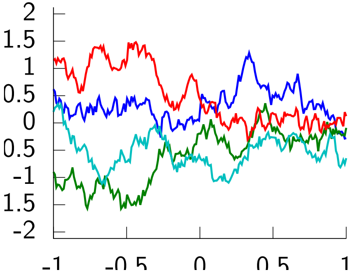
Covariance functions

Where did this covariance matrix come from?

Ornstein-Uhlenbeck (stationary Gauss-Markov) covariance function

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

- One dimension: arises from a stochastic differential equation (Brownian motion in a parabolic tube).
- Higher dimensions: Fourier filter of the form $\frac{1}{\pi(1+x^2)}$.



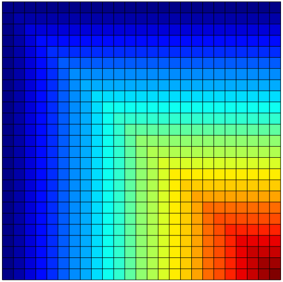
Covariance functions

Where did this covariance matrix come from?

Markov process

$$k\left(t, t'\right) = \alpha \min(t, t')$$

- Covariance matrix is built using the *inputs* to the function t.



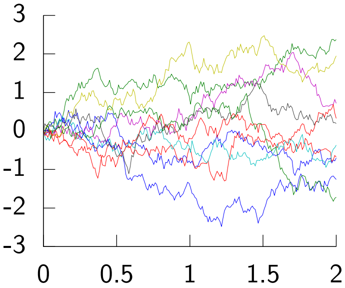
Covariance functions

Where did this covariance matrix come from?

Markov process

$$k\left(t, t'\right) = \alpha \min(t, t')$$

- Covariance matrix is built using the *inputs* to the function t.



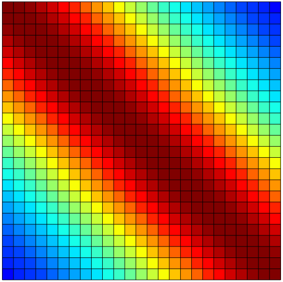
Covariance functions

Where did this covariance matrix come from?

Matern 5/2 covariance function

$$k\left(\mathbf{x}, \mathbf{x}'\right) = \alpha\left(1 + \sqrt{5}r + \frac{5}{3}r^2\right)\exp\left(-\sqrt{5}r\right) \quad \text{where} \quad r = \frac{\left\|\mathbf{x} - \mathbf{x}'\right\|_2}{\ell}$$

- Matern 5/2 is a twice differentiable covariance.
- Matern family constructed with Student-*t* filters in Fourier space.



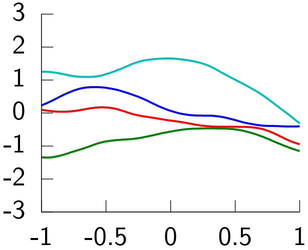
Covariance functions

Where did this covariance matrix come from?

Matern 5/2 covariance function

$$k\left(\mathbf{x}, \mathbf{x}'\right) = \alpha \left(1 + \sqrt{5} r + \frac{5}{3} r^2\right) \exp\left(-\sqrt{5} r\right) \quad \text{where} \quad r = \frac{\left\|\mathbf{x} - \mathbf{x}'\right\|_2}{\ell}$$

- Matern 5/2 is a twice differentiable covariance.
- Matern family constructed with Student-t filters in Fourier space.



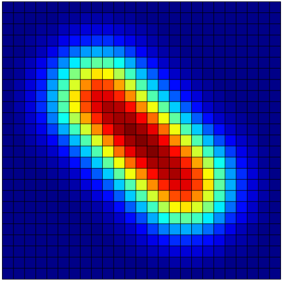
Covariance functions

RBF basis functions

$$k\left(\mathbf{x}, \mathbf{x}'\right) = \alpha \boldsymbol{\phi}(\mathbf{x})^\top \boldsymbol{\phi}(\mathbf{x}')$$

$$\phi_k(x) = \exp\left(-\frac{\left\|x - \mu_k\right\|_2^2}{\ell^2}\right)$$

$$\boldsymbol{\mu} = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$



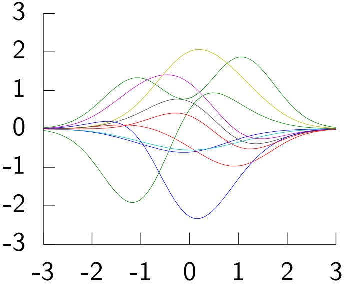
Covariance functions

RBF basis functions

$$k\left(\mathbf{x}, \mathbf{x}'\right) = \alpha \boldsymbol{\phi}(\mathbf{x})^\top \boldsymbol{\phi}(\mathbf{x}')$$

$$\phi_k(x) = \exp\left(-\frac{\left\|x - \mu_k\right\|_2^2}{\ell^2}\right)$$

$$\boldsymbol{\mu} = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$



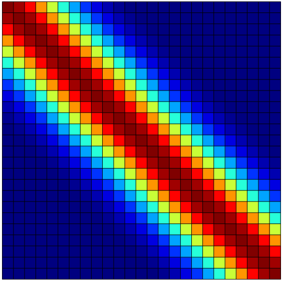
Covariance functions

Where did this covariance matrix come from?

Exponentiated quadratic kernel function (RBF, squared exponential, Gaussian)

$$k(\mathbf{x}, \mathbf{x}') = \alpha \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|_2^2}{2\ell^2}\right)$$

- Covariance matrix built using the *inputs* to the function **x**.
- Example above: based on Euclidean distance.
- The covariance function a.k.a kernel.



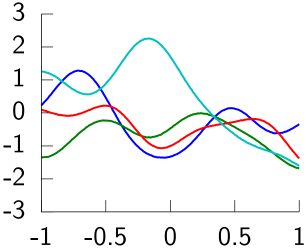
Covariance functions

Where did this covariance matrix come from?

Exponentiated quadratic kernel function (RBF, squared exponential, Gaussian)

$$k\left(\mathbf{x}, \mathbf{x}'\right) = \alpha \exp\left(-\frac{\left\|\mathbf{x} - \mathbf{x}'\right\|_2^2}{2\ell^2}\right)$$

- Covariance matrix built using the *inputs* to the function $\mathbf{x}$.
- Example above: based on Euclidean distance.
- The covariance function a.k.a kernel.



1. **Probability helps to model any source of uncertainty.**

2. **The Gaussian appear in many operations in science.**

3. **Correlations are key in the samples from the Gaussian.**

4. **Gaussian processes are distributions over functions.**

Constructing covariance functions

- Sum of two covariances is also a covariance function.

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

Constructing covariance functions

- Product of two covariances is also a covariance function.

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

Multiply by deterministic function

- If $f(\mathbf{x})$ is a Gaussian process.
- $g(\mathbf{x})$ is a deterministic function.
- $h(\mathbf{x}) = f(\mathbf{x})g(\mathbf{x})$
- Then
  $$k_h(\mathbf{x}, \mathbf{x}') = g(\mathbf{x})k_f(\mathbf{x}, \mathbf{x}')g(\mathbf{x}')$$
  where k_h is covariance for $h(\cdot)$ and k_f is covariance for $f(\cdot)$.

1. **Probability helps to model any source of uncertainty.**
2. **The Gaussian appear in many operations in science.**
3. **Correlations are key in the samples from the Gaussian.**
4. **Gaussian processes are distributions over functions.**
5. **New covariances can be derived from old ones.**

Prediction with correlated Gaussians

$$y_i = f(\mathbf{x}_i)$$

- Training input and output pairs $(\mathbf{X}, \mathbf{y})$, and test inputs $\mathbf{x}_*$.
- Prediction of $\mathbf{f}_*$ from $\mathbf{f}$: multivariate *conditional density*.
- Here the joint density is given by $p(\mathbf{f}, \mathbf{f}_*|\mathbf{X}) \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{K}\right)$

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{\mathbf{f},\mathbf{f}} & \mathbf{K}_{*,\mathbf{f}} \\ \mathbf{K}_{\mathbf{f},*} & \mathbf{K}_{*,*} \end{bmatrix}$$

- Multivariate conditional density is *also* Gaussian.

$$p(\mathbf{f}_*|\mathbf{f}) = \mathcal{N}\left(\mathbf{f}_*|\mathbf{K}_{*,\mathbf{f}}\mathbf{K}_{\mathbf{f},\mathbf{f}}^{-1}\mathbf{f}, \mathbf{K}_{*,*} - \mathbf{K}_{*,\mathbf{f}}\mathbf{K}_{\mathbf{f},\mathbf{f}}^{-1}\mathbf{K}_{\mathbf{f},*}\right)$$

Prediction with correlated Gaussians

$$y_i = f(\mathbf{x}_i)$$

- Training input and output pairs $(\mathbf{X}, \mathbf{y})$, and test inputs $\mathbf{x}_*$.
- Prediction of $\mathbf{f}_*$ from $\mathbf{f}$: multivariate *conditional density*.
- Here the joint density is given by $p(\mathbf{f}, \mathbf{f}_* | \mathbf{X}) \sim \mathcal{N}(\mathbf{0}, \boldsymbol{K})$

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{\mathbf{f},\mathbf{f}} & \mathbf{K}_{*,\mathbf{f}} \\ \mathbf{K}_{\mathbf{f},*} & \mathbf{K}_{*,*} \end{bmatrix}$$

- Multivariate conditional density is *also* Gaussian.

$$p(\mathbf{f}_* | \mathbf{f}) = \mathcal{N}(\mathbf{f}_* | \boldsymbol{\mu}, \boldsymbol{\Sigma})$$

$$\boldsymbol{\mu} = \mathbf{K}_{*,\mathbf{f}} \mathbf{K}_{\mathbf{f},\mathbf{f}}^{-1} \mathbf{f}$$

$$\boldsymbol{\Sigma} = \mathbf{K}_{*,*} - \mathbf{K}_{*,\mathbf{f}} \mathbf{K}_{\mathbf{f},\mathbf{f}}^{-1} \mathbf{K}_{\mathbf{f},*}$$

Gaussian process interpolation

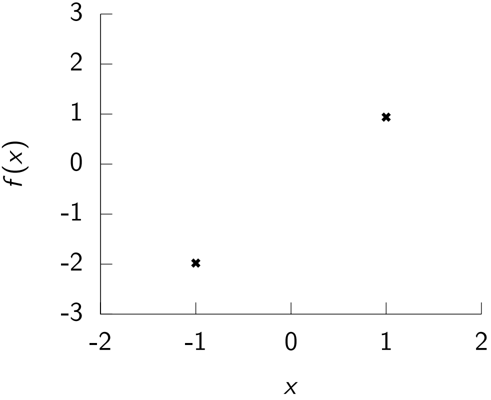


Figure: Interpolation through outputs from slow computer simulations (*e.g.* atmospheric carbon levels).

Gaussian process interpolation

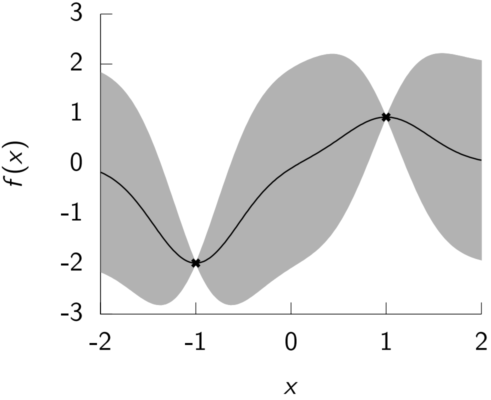


Figure: Interpolation through outputs from slow computer simulations (*e.g.* atmospheric carbon levels).

Gaussian process interpolation

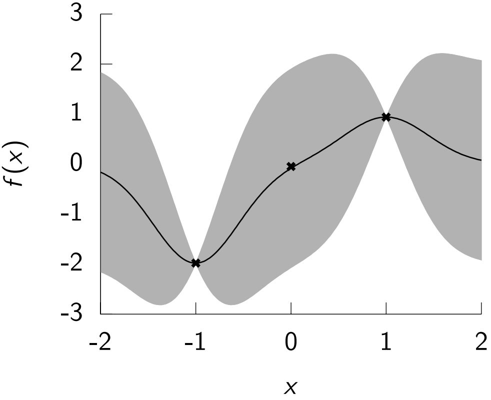


Figure: Interpolation through outputs from slow computer simulations (*e.g.* atmospheric carbon levels).

Gaussian process interpolation

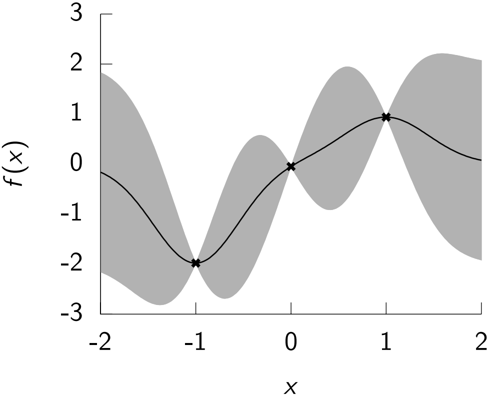


Figure: Interpolation through outputs from slow computer simulations (*e.g.* atmospheric carbon levels).

Gaussian process interpolation

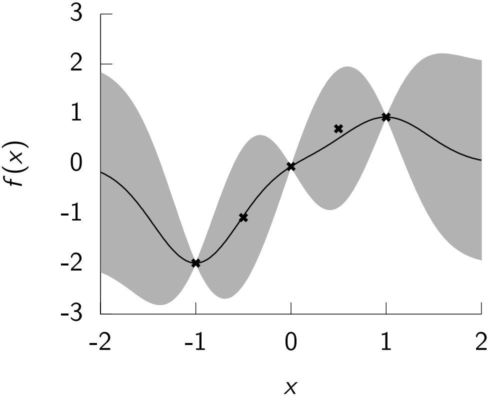


Figure: Interpolation through outputs from slow computer simulations (*e.g.* atmospheric carbon levels).

Gaussian process interpolation

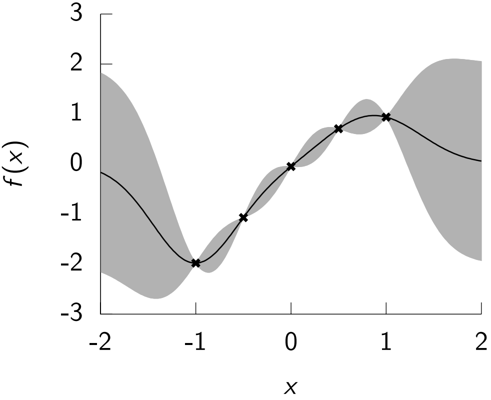


Figure: Interpolation through outputs from slow computer simulations (*e.g.* atmospheric carbon levels).

Gaussian process interpolation

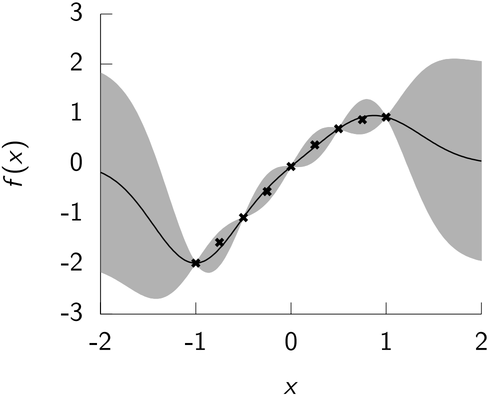


Figure: Interpolation through outputs from slow computer simulations (*e.g.* atmospheric carbon levels).

Gaussian process interpolation

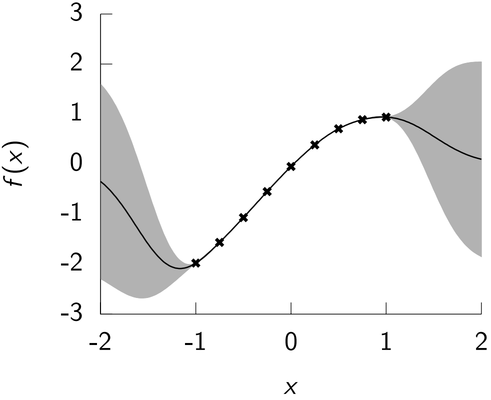


Figure: Interpolation through outputs from slow computer simulations (*e.g.* atmospheric carbon levels).

Prediction with noisy observations

$$y_i = f(\mathbf{x}_i) + \epsilon_i$$

- Gaussian noise model,

  $$p(y_i|f_i) = \mathcal{N}\left(y_i|f_i, \sigma^2\right)$$

  where σ^2 is the variance of the noise.

- Equivalent to a covariance function of the form

  $$k(\mathbf{x}_i, \mathbf{x}_j) = \delta_{i,j}\sigma^2$$

  where $\delta_{i,j}$ is the Kronecker delta function.

- Additive nature of Gaussians means we can simply add this term to existing covariance matrices.

  $$Cov(y_i, y_j) = k(\mathbf{x}_i, \mathbf{x}_j) + \delta_{i,j}\sigma^2$$

Prediction with noisy observations

- Training input and output pairs ($\mathbf{X}$, $\mathbf{y}$), and test inputs $\mathbf{x}_*$.
- Distribution of the observed target values and the function values at the test location $p(\mathbf{y}, \mathbf{f}_*) \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{K}\right)$

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{\mathbf{f},\mathbf{f}} + \sigma^2 \mathbf{I} & \mathbf{K}_{*,\mathbf{f}} \\ \mathbf{K}_{\mathbf{f},*} & \mathbf{K}_{*,*} \end{bmatrix}$$

- Multivariate conditional density is *also* Gaussian.

$$p(\mathbf{f}_* | \mathbf{X}, \mathbf{y}, \mathbf{x}_*) = \mathcal{N}\left(\mathbf{f}_* | \mathbf{K}_{*,\mathbf{f}} \tilde{\mathbf{K}}_{\mathbf{f},\mathbf{f}} \mathbf{f}, \mathbf{K}_{*,*}^{-1} - \tilde{\mathbf{K}}_{\mathbf{f},\mathbf{f}} \mathbf{K}_{\mathbf{f},*}\right)$$

$$\tilde{\mathbf{K}}_{\mathbf{f},\mathbf{f}} = \mathbf{K}_{\mathbf{f},\mathbf{f}} + \sigma^2 \mathbf{I}$$

Prediction with noisy observations

- Training input and output pairs $(\mathbf{X}, \mathbf{y})$, and test inputs $\mathbf{x}_*$.
- Distribution of the observed target values and the function values at the test location $p(\mathbf{y}, \mathbf{f}_*) \sim \mathcal{N}(\mathbf{0}, \boldsymbol{K})$

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{\mathbf{f},\mathbf{f}} + \sigma^2 \mathbf{I} & \mathbf{K}_{*,\mathbf{f}} \\ \mathbf{K}_{\mathbf{f},*} & \mathbf{K}_{*,*} \end{bmatrix}$$

- Multivariate conditional density is *also* Gaussian.

$$p(\mathbf{f}_* | \mathbf{X}, \mathbf{y}, \mathbf{x}_*) = \mathcal{N}(\mathbf{f}_* | \boldsymbol{\mu}, \Sigma)$$

$$\tilde{\mathbf{K}}_{\mathbf{f},\mathbf{f}} = \mathbf{K}_{\mathbf{f},\mathbf{f}} + \sigma^2 \mathbf{I}$$

$$\boldsymbol{\mu} = \mathbf{K}_{*,\mathbf{f}} \tilde{\mathbf{K}}_{\mathbf{f},\mathbf{f}}^{-1} \mathbf{f}$$

$$\Sigma = \mathbf{K}_{*,*} - \mathbf{K}_{*,\mathbf{f}} \tilde{\mathbf{K}}_{\mathbf{f},\mathbf{f}}^{-1} \mathbf{K}_{\mathbf{f},*}$$

Predictions for y_* with Gaussian likelihoods

Training input and output pairs $(\mathbf{X}, \mathbf{y})$, and test input $\mathbf{x}_*$.

Model:

$$y_i = f(\mathbf{x}_i) + \epsilon_i$$

$$f \sim \mathcal{GP}(f|0, K)$$

$$\epsilon_i \sim \mathcal{N}(\epsilon_i|0, \sigma^2)$$

Likelihood:

$$p(\mathbf{y}|\mathbf{X}, f) = \prod_{i=1}^{n} p(y_i|f, \mathbf{x}_i)$$

Predictions:

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

Predictions for y_* with Gaussian likelihoods

Training input and output pairs $(\mathbf{X}, \mathbf{y})$, and test input $\mathbf{x}_*$.

Model:

$$y_i = f(\mathbf{x}_i) + \epsilon_i$$

$$f \sim \mathcal{GP}(f|0, K)$$

$$\epsilon_i \sim \mathcal{N}(\epsilon_i|0, \sigma^2)$$

Likelihood:

$$p(\mathbf{y}|\mathbf{X}, f) = \prod_{i=1}^{n} p(y_i|f, \mathbf{x}_i)$$

Predictions:

$$p(y_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y}) = \mathcal{N}\left(y_*|\boldsymbol{\mu}, \boldsymbol{\Sigma} + \sigma^2\right)$$

Gaussian process regression

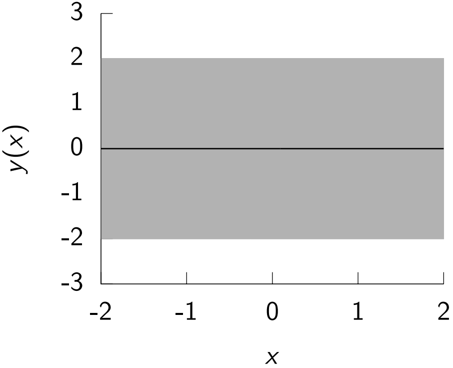


Figure: Fitting through outputs (with noise) from slow computer simulations (*e.g.* atmospheric carbon levels).

Gaussian process regression

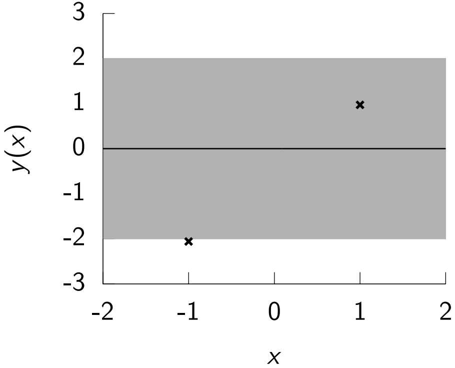


Figure: Fitting through outputs (with noise) from slow computer simulations (*e.g.* atmospheric carbon levels).

Gaussian process regression

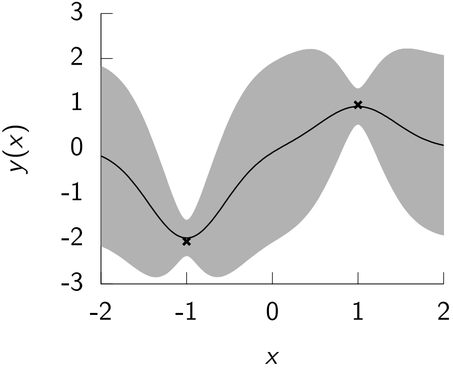


Figure: Fitting through outputs (with noise) from slow computer simulations (*e.g.* atmospheric carbon levels).

Gaussian process regression

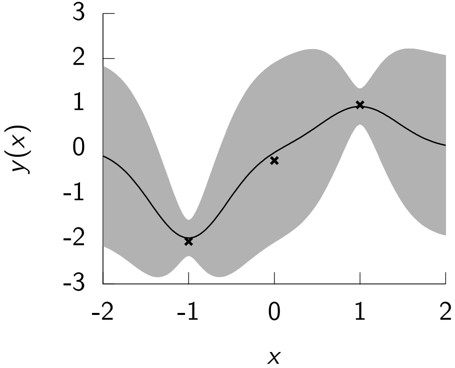


Figure: Fitting through outputs (with noise) from slow computer simulations (*e.g.* atmospheric carbon levels).

Gaussian process regression

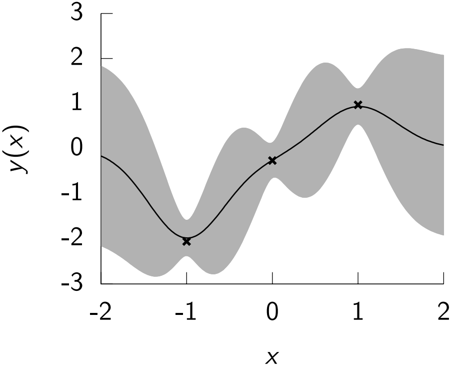


Figure: Fitting through outputs (with noise) from slow computer simulations (*e.g.* atmospheric carbon levels).

Gaussian process regression

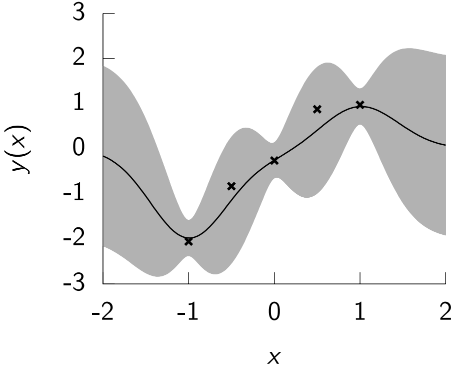


Figure: Fitting through outputs (with noise) from slow computer simulations (*e.g.* atmospheric carbon levels).

Gaussian process regression

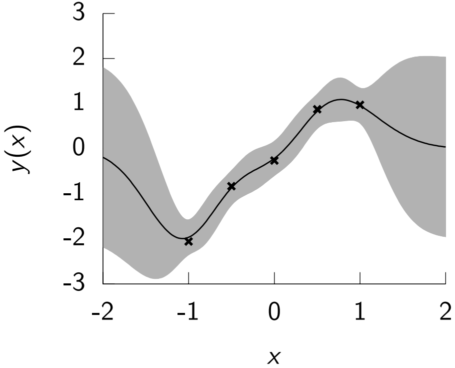


Figure: Fitting through outputs (with noise) from slow computer simulations (*e.g.* atmospheric carbon levels).

Gaussian process regression

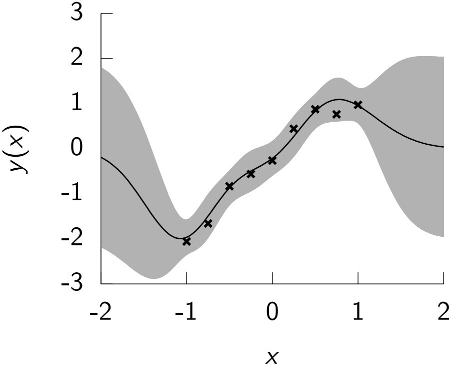


Figure: Fitting through outputs (with noise) from slow computer simulations (*e.g.* atmospheric carbon levels).

Gaussian process regression

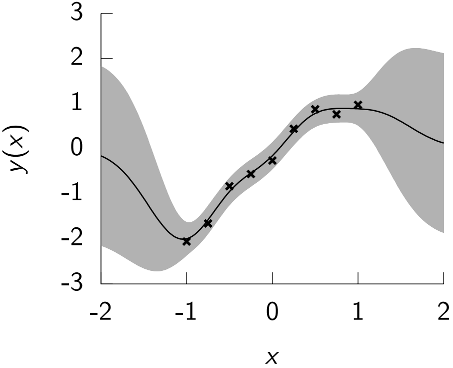


Figure: Fitting through outputs (with noise) from slow computer simulations (*e.g.* atmospheric carbon levels).

Considerations

- GPs allows to characterize and quantify uncertainty about f.
- The mean of the GP is a linear predictor: $\boldsymbol{\mu} = \mathbf{K}_{*,\mathbf{f}}\alpha$.
- The inversion of $\tilde{\mathbf{K}}_{\mathbf{f},\mathbf{f}} = \mathbf{K}_{\mathbf{f},\mathbf{f}} + \sigma^2\mathbf{I}$ costs $\mathcal{O}(N^3)$.
- The prediction cost per new point is $\mathcal{O}(N^2)$.

1. **Probability helps to model any source of uncertainty.**
2. **The Gaussian appear in many operations in science.**
3. **Correlations are key in the samples from the Gaussian.**
4. **Gaussian processes are distributions over functions.**
5. **New covariances can be derived from old ones.**
6. **Prediction in GPs are linear algebra operations.**

Questions at this point

- Given a covariance (prior), how to select the right parameters?

- How to deal with non Gaussian likelihoods?

- How is this representation of the GP related to a basis function representation (and therefore conected to splines, NNs, etc)?

Learning covariance parameters

Can we determine covariance parameters from the data?

- Advantage of GPs: hyperparameters and covariances can be chonse directly from the training data (no cross validation).

- Minimize the negative log-marginal likelihood $\mathcal{L}(\theta)$ w.r.t kernel hyper paremeters and noise, $\mathbf{K}(\theta) = \mathbf{K} + \sigma^2\mathbf{I}$.

$$p(\mathbf{y}|\mathbf{X}) = \int p(\mathbf{y}|f, \mathbf{X})p(f)df \sim \mathcal{N}\left(\mathbf{y}|\mathbf{0}, \mathbf{K}(\theta)\right)$$

$$\mathcal{N}\left(\mathbf{y}|\mathbf{0}, \mathbf{K}(\theta)\right) = \frac{1}{(2\pi)^{\frac{n}{2}}|\mathbf{K}(\theta)|^{\frac{1}{2}}}\exp\left(-\frac{\mathbf{y}^\top\mathbf{K}(\theta)^{-1}\mathbf{y}}{2}\right)$$

Learning covariance parameters

Can we determine covariance parameters from the data?

- Advantage of GPs: hyperparameters and covariances can be chonse directly from the training data (no cross validation).

- Minimize the negative log-marginal likelihood $\mathcal{L}(\theta)$ w.r.t kernel hyper paremeters and noise, $\mathbf{K}(\theta) = \mathbf{K} + \sigma^2\mathbf{I}$.

$$p(\mathbf{y}|\mathbf{X}) = \int p(\mathbf{y}|f, \mathbf{X})p(f)df \sim \mathcal{N}\left(\mathbf{y}|\mathbf{0}, \mathbf{K}(\theta)\right)$$

$$\mathcal{N}\left(\mathbf{y}|\mathbf{0}, \mathbf{K}(\theta)\right) = \frac{1}{(2\pi)^{\frac{n}{2}}|\mathbf{K}(\theta)|^{\frac{1}{2}}}\exp\left(-\frac{\mathbf{y}^\top \mathbf{K}(\theta)^{-1}\mathbf{y}}{2}\right)$$

Learning covariance parameters

Can we determine covariance parameters from the data?

- Advantage of GPs: hyperparameters and covariances can be chonse directly from the training data (no cross validation).

- Minimize the negative log-marginal likelihood $\mathcal{L}(\theta)$ w.r.t kernel hyper paremeters and noise, $\mathbf{K}(\theta) = \mathbf{K} + \sigma^2 \mathbf{I}$.

$$p(\mathbf{y}|\mathbf{X}) = \int p(\mathbf{y}|f, \mathbf{X}) p(f) df \sim \mathcal{N}\left(\mathbf{y}|\mathbf{0}, \mathbf{K}(\theta)\right)$$

$$\log \mathcal{N}\left(\mathbf{y}|\mathbf{0}, \mathbf{K}(\theta)\right) = -\frac{1}{2} \log |\mathbf{K}(\theta)| - \frac{\mathbf{y}^\top \mathbf{K}(\theta)^{-1} \mathbf{y}}{2} - \frac{n}{2} \log 2\pi$$

Learning covariance parameters

Can we determine covariance parameters from the data?

- Advantage of GPs: hyperparameters and covariances can be chonse directly from the training data (no cross validation).

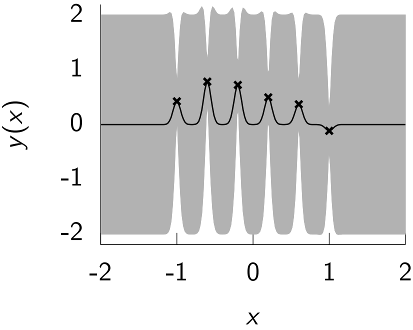
- Minimize the negative log-marginal likelihood $\mathcal{L}(\theta)$ w.r.t kernel hyper paremeters and noise, $\mathbf{K}(\theta) = \mathbf{K} + \sigma^2 \mathbf{I}$.

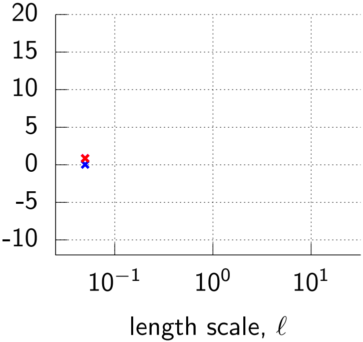
$$p(\mathbf{y}|\mathbf{X}) = \int p(\mathbf{y}|f, \mathbf{X}) p(f) df \sim \mathcal{N}\left(\mathbf{y}|\mathbf{0}, \mathbf{K}(\theta)\right)$$

$$\mathcal{L}(\theta) = -\log \mathcal{N}\left(\mathbf{y}|\mathbf{0}, \mathbf{K}(\theta)\right) = \frac{1}{2} \log |\mathbf{K}(\theta)| + \frac{\mathbf{y}^\top \mathbf{K}(\theta)^{-1} \mathbf{y}}{2} + \frac{n}{2} \log 2\pi$$

Learning covariance parameters

Can we determine length scales and noise levels from the data?

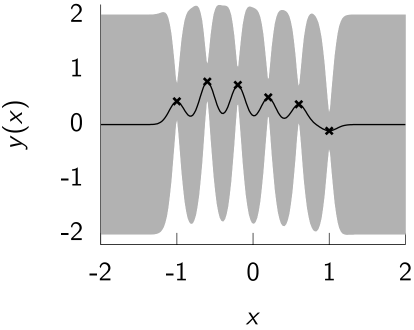


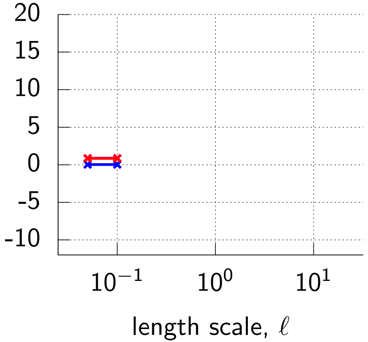


$$\mathcal{L}(\theta) = \frac{1}{2}\log|\mathbf{K}(\theta)| + \frac{\mathbf{y}^\top \mathbf{K}(\theta)^{-1}\mathbf{y}}{2} + \frac{n}{2}\log 2\pi$$

Learning covariance parameters

Can we determine length scales and noise levels from the data?

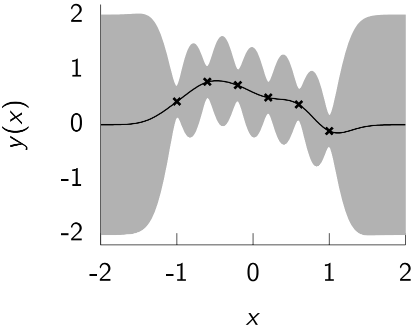


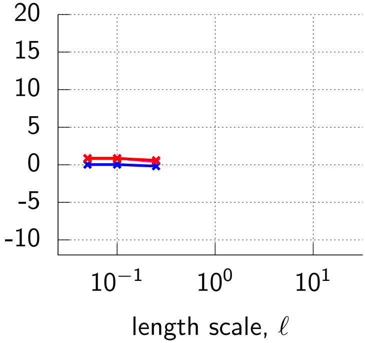


$$\mathcal{L}(\theta) = \frac{1}{2}\log|\mathbf{K}(\theta)| + \frac{\mathbf{y}^\top \mathbf{K}(\theta)^{-1}\mathbf{y}}{2} + \frac{n}{2}\log 2\pi$$

Learning covariance parameters

Can we determine length scales and noise levels from the data?

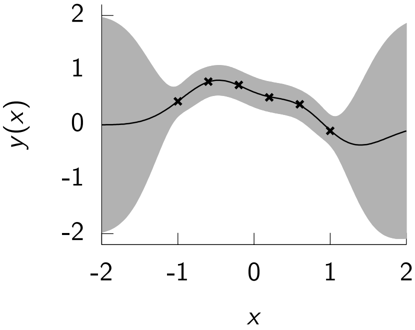


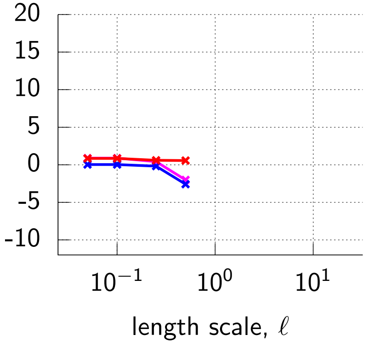


$$\mathcal{L}(\theta) = \frac{1}{2}\log|\mathbf{K}(\theta)| + \frac{\mathbf{y}^\top\mathbf{K}(\theta)^{-1}\mathbf{y}}{2} + \frac{n}{2}\log 2\pi$$

Learning covariance parameters

Can we determine length scales and noise levels from the data?

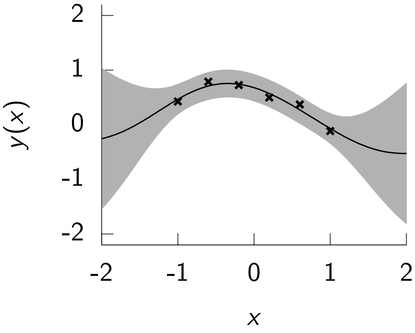


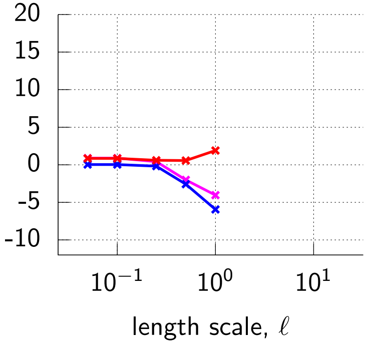


$$\mathcal{L}(\theta) = \frac{1}{2}\log|\mathbf{K}(\theta)| + \frac{\mathbf{y}^\top \mathbf{K}(\theta)^{-1}\mathbf{y}}{2} + \frac{n}{2}\log 2\pi$$

Learning covariance parameters

Can we determine length scales and noise levels from the data?

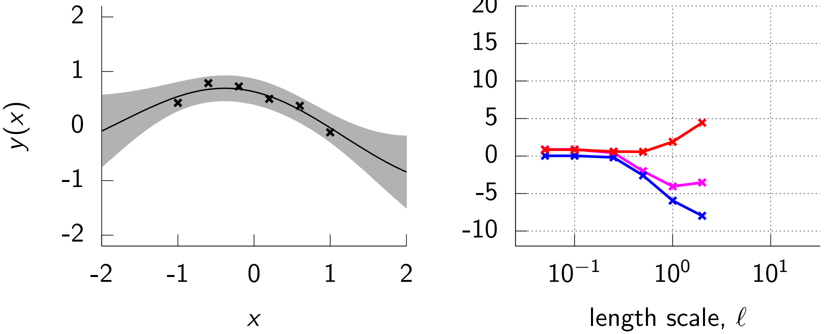




$$\mathcal{L}(\theta) = \frac{1}{2}\log|\mathbf{K}(\theta)| + \frac{\mathbf{y}^\top \mathbf{K}(\theta)^{-1}\mathbf{y}}{2} + \frac{n}{2}\log 2\pi$$

Learning covariance parameters

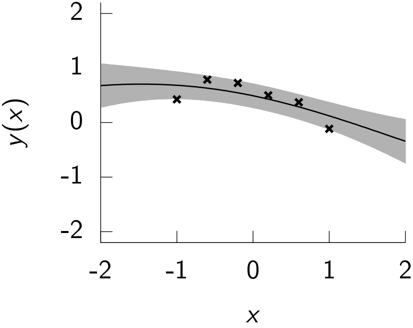
Can we determine length scales and noise levels from the data?

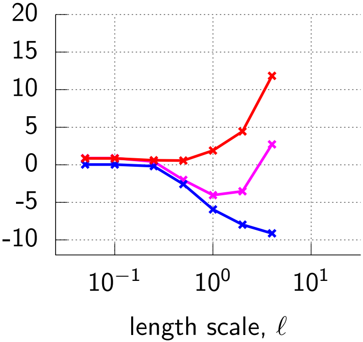


$$\mathcal{L}(\theta) = \frac{1}{2}\log|\mathbf{K}(\theta)| + \frac{\mathbf{y}^{\top}\mathbf{K}(\theta)^{-1}\mathbf{y}}{2} + \frac{n}{2}\log 2\pi$$

Learning covariance parameters

Can we determine length scales and noise levels from the data?

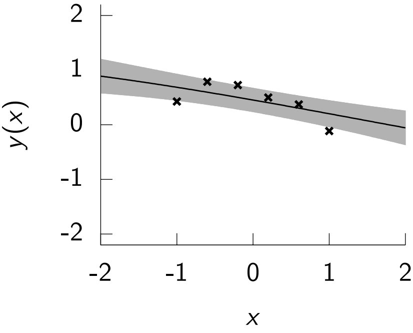


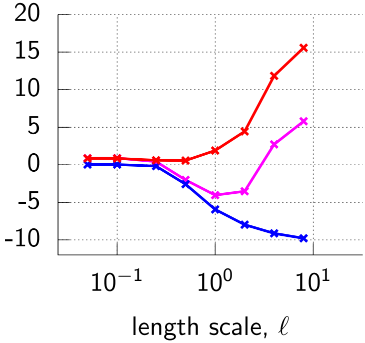


$$\mathcal{L}(\theta) = \frac{1}{2}\log|\mathbf{K}(\theta)| + \frac{\mathbf{y}^\top\mathbf{K}(\theta)^{-1}\mathbf{y}}{2} + \frac{n}{2}\log 2\pi$$

Learning covariance parameters

Can we determine length scales and noise levels from the data?

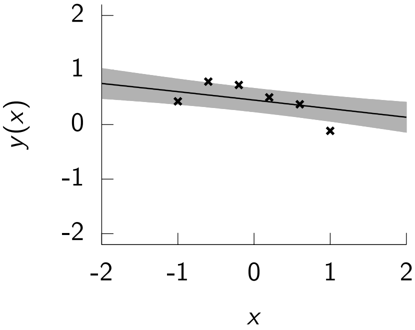


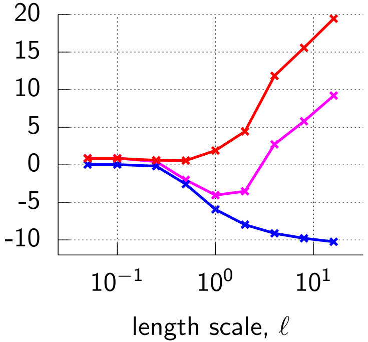


$$\mathcal{L}(\theta) = \frac{1}{2}\log|\mathbf{K}(\theta)| + \frac{\mathbf{y}^\top \mathbf{K}(\theta)^{-1}\mathbf{y}}{2} + \frac{n}{2}\log 2\pi$$

Learning covariance parameters

Can we determine length scales and noise levels from the data?





$$\mathcal{L}(\theta) = \frac{1}{2}\log|\mathbf{K}(\theta)| + \frac{\mathbf{y}^\top \mathbf{K}(\theta)^{-1}\mathbf{y}}{2} + \frac{n}{2}\log 2\pi$$

1. **Probability helps to model any source of uncertainty.**
2. **The Gaussian appear in many operations in science.**
3. **Correlations are key in the samples from the Gaussian.**
4. **Gaussian processes are distributions over functions.**
5. **New covariances can be derived from old ones.**
6. **Prediction in GPs are linear algebra operations.**
7. **Marginal likelihood as objective for training. No CV.**

Non Gaussian likelihoods

Gaussian processes for classification as example

Training input and output pairs $(\mathbf{X}, \mathbf{y})$, where $y_i \pm 1$.

Model:

- Sigmoidal likelihood: $p(y = +1|f, \mathbf{x}) = \sigma(f(\mathbf{x}))$.
- Prior over f: $f \sim \mathcal{GP}(f|0, K)$.

Predictive for f_*:

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

where $p(\mathbf{f}|\mathbf{X}, \mathbf{y}) = p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\mathbf{X})/p(\mathbf{y}|\mathbf{X})$ is the posterior over f.

Predictive for y_*:

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

Non Gaussian likelihoods

Gaussian processes for classification as example

Training input and output pairs $(\mathbf{X}, \mathbf{y})$, where $y_i \pm 1$.

Model:

- Sigmoidal likelihood: $p(y = +1|f, \mathbf{x}) = \sigma(f(\mathbf{x}))$.
- Prior over f: $f \sim \mathcal{GP}(f|0, K)$.

Predictive for f_*:

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

where $p(\mathbf{f}|\mathbf{X}, \mathbf{y}) = p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\mathbf{X})/p(\mathbf{y}|\mathbf{X})$ is the posterior over f.

Predictive for y_*:

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

Non Gaussian likelihoods

Gaussian processes for classification as example

Training input and output pairs $(\mathbf{X}, \mathbf{y})$, where $y_i \pm 1$.

Model:

- Sigmoidal likelihood: $p(y = +1|f, \mathbf{x}) = \sigma(f(\mathbf{x}))$.
- Prior over f: $f \sim \mathcal{GP}(f|0, K)$.

Predictive for f_*:

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

where $p(\mathbf{f}|\mathbf{X}, \mathbf{y}) = p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\mathbf{X})/p(\mathbf{y}|\mathbf{X})$ is the posterior over f.

Predictive for y_*:

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

Non Gaussian likelihoods

Gaussian processes for classification as example

Training input and output pairs $(\mathbf{X}, \mathbf{y})$, where $y_i \pm 1$.

Model:

- Sigmoidal likelihood: $p(y = +1|f, \mathbf{x}) = \sigma(f(\mathbf{x}))$.
- Prior over f: $f \sim \mathcal{GP}(f|0, K)$.

Predictive for f_*:

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

where $p(\mathbf{f}|\mathbf{X}, \mathbf{y}) = p(\mathbf{y}|\mathbf{f}) p(\mathbf{f}|\mathbf{X}) / p(\mathbf{y}|\mathbf{X})$ is the posterior over f.

Predictive for y_*:

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

Approximations

$$p(\mathbf{f}|\mathbf{X},\mathbf{y}) = p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\mathbf{X})/p(\mathbf{y}|\mathbf{X}) \text{ is intractable.}$$

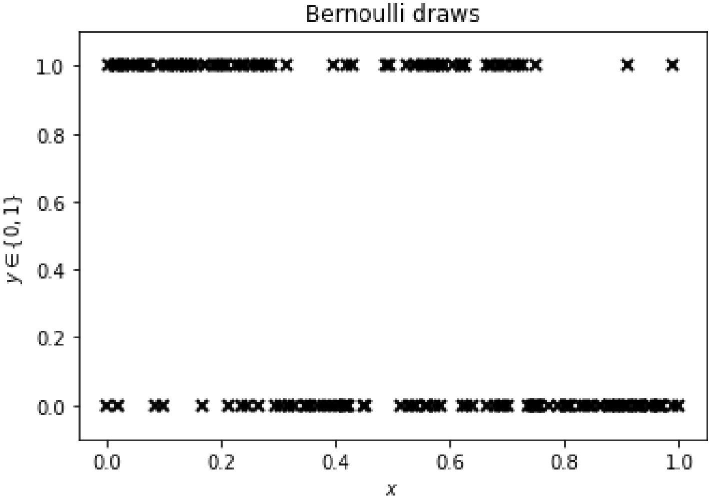
Two common ways to make Gaussian approximation to posterior:

- **Laplace approximation**: Uses the second order Taylor approximation at the mode of the posterior.

- **Expectation propagation**: Can be studied as of as approximately minimizing $KL[p(\mathbf{f}|\mathbf{X},\mathbf{y})||q(\mathbf{f}|\mathbf{X},\mathbf{y})]$ by an iterative procedure for some simple form of $q(\mathbf{f}|\mathbf{X},\mathbf{y})$.

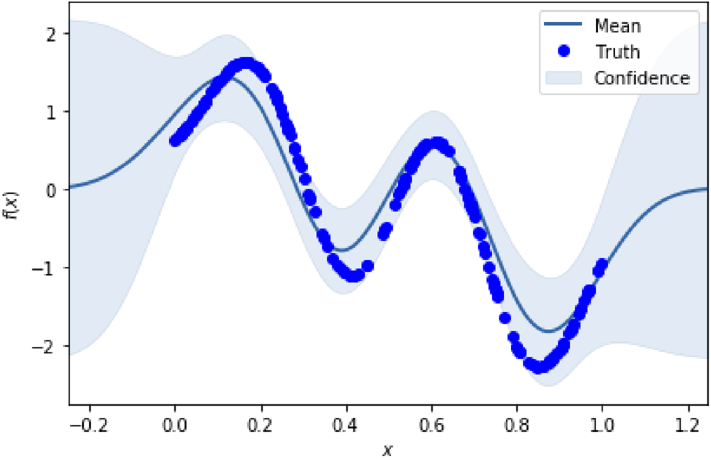
Example

Data from two classes



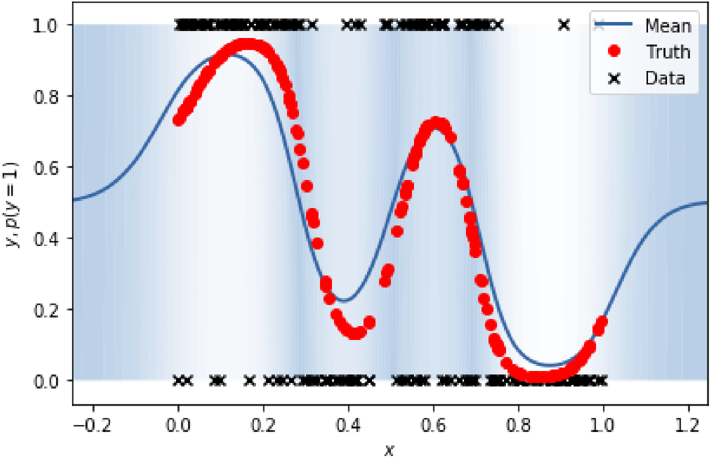
Example

Latent function after optimization



Example

Warped latent function after optimization



1. **Probability helps to model any source of uncertainty.**
2. **The Gaussian appear in many operations in science.**
3. **Correlations are key in the samples from the Gaussian.**
4. **Gaussian processes are distributions over functions.**
5. **New covariances can be derived from old ones.**
6. **Prediction in GPs are linear algebra operations.**
7. **Marginal likelihood as objective for training. No CV.**
8. **Non Gaussian likelihoods require approximations.**

GPs: basis of functions point of view

- GPs can be seen as a generalization of Bayesian linear regression with infinite number of basis.

- Next we review this connection.

Basis function representations

- Represent a function by a linear sum over a basis,

$$f(\mathbf{x}_i; \mathbf{w}) = \sum_{k=1}^{m} w_k \phi_k(\mathbf{x}_i),$$

.

- Here: m basis functions and $\phi_k(\cdot)$ is kth basis function and

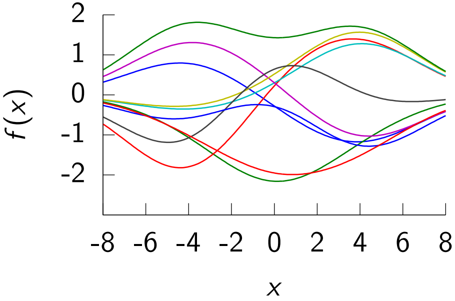
$$\mathbf{w} = [w_1, \ldots, w_m]^\top .$$

- For standard linear model: $\phi_k(\mathbf{x}_i) = \mathbf{x}_i$.

Random functions

Functions derived using $f(x) = \sum_{k=1}^{m} w_k \phi_k(x)$ where elements of $\mathbf{w}$ are independently sampled from a Gaussian density,

$$w_k \sim \mathcal{N}(0, \alpha).$$



Direct construction of covariance matrix

Use matrix notation to write function,

$$f\left(\mathbf{x}_i; \mathbf{w}\right) = \sum_{k=1}^{m} w_k \phi_k\left(\mathbf{x}_i\right).$$

Direct construction of covariance matrix

Use matrix notation to write function,

$$f\left(\mathbf{x}_i; \mathbf{w}\right) = \sum_{k=1}^{m} w_k \phi_k\left(\mathbf{x}_i\right).$$

- Computed at training data gives a vector $\mathbf{f} = \boldsymbol{\Phi}\mathbf{w}$.
- $\mathbf{w}$ and $\mathbf{f}$ are only related by an *inner product*.
- $\boldsymbol{\Phi} \in \Re^{n\times p}$ is a *design matrix* (fixed and non-stochastic for a given training set).

Expectations

- We have
$$\mathbb{E}[\mathbf{f}] = \mathbf{\Phi}\mathbb{E}[\mathbf{w}].$$

- Prior mean of $\mathbf{w}$ was zero giving
$$\mathbb{E}[\mathbf{f}] = \mathbf{0}.$$

- Prior covariance of $\mathbf{f}$ is
$$\mathbf{K} = \mathbb{E}[\mathbf{f}\mathbf{f}^\top] - \mathbb{E}[\mathbf{f}]\mathbb{E}[\mathbf{f}]^\top = \mathbb{E}[\mathbf{f}\mathbf{f}^\top]$$

Expectations

- We have
$$\mathbb{E}[\mathbf{f}] = \mathbf{\Phi}\mathbb{E}[\mathbf{w}].$$
- Prior mean of **w** was zero giving
$$\mathbb{E}[\mathbf{f}] = \mathbf{0}.$$
- Prior covariance of **f** is
$$\mathbf{K} = \mathbb{E}[\mathbf{f}\mathbf{f}^\top] - \mathbb{E}[\mathbf{f}]\mathbb{E}[\mathbf{f}]^\top = \mathbb{E}[\mathbf{f}\mathbf{f}^\top]$$

Expectations

- We have

$$\mathbb{E}[\mathbf{f}] = \boldsymbol{\Phi}\mathbb{E}[\mathbf{w}].$$

- Prior mean of **w** was zero giving

$$\mathbb{E}[\mathbf{f}] = \mathbf{0}.$$

- Prior covariance of **f** is

$$\mathbf{K} = \mathbb{E}[\mathbf{f}\mathbf{f}^\top] - \mathbb{E}[\mathbf{f}]\mathbb{E}[\mathbf{f}]^\top = \mathbb{E}[\mathbf{f}\mathbf{f}^\top]$$

Expectations

- We have

$$\mathbb{E}[\mathbf{f}] = \boldsymbol{\Phi}\mathbb{E}[\mathbf{w}].$$

- Prior mean of $\mathbf{w}$ was zero giving

$$\mathbb{E}[\mathbf{f}] = \mathbf{0}.$$

- Prior covariance of $\mathbf{f}$ is

$$\mathbf{K} = \mathbb{E}[\mathbf{f}\mathbf{f}^\top] - \mathbb{E}[\mathbf{f}]\mathbb{E}[\mathbf{f}]^\top = \mathbb{E}[\mathbf{f}\mathbf{f}^\top]$$

$$\mathbb{E}[\mathbf{f}\mathbf{f}^\top] = \boldsymbol{\Phi}\mathbb{E}[\mathbf{w}\mathbf{w}^\top]\boldsymbol{\Phi}^\top,$$

giving

$$\mathbf{K} = \alpha\boldsymbol{\Phi}\boldsymbol{\Phi}^\top.$$

Radial basis functions

Basis function maps data into a "feature space" in which a linear sum is a non linear function

$$\phi_k\left(\mathbf{x}_i\right) = \exp\left(-\frac{\left|\mathbf{x}_i - \boldsymbol{\mu}_k\right|^2}{2\ell^2}\right).$$

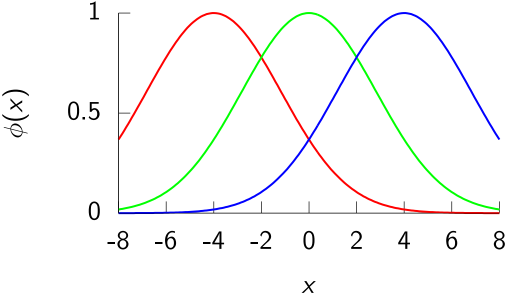


Figure: A set of radial basis functions with width $\ell = 2$ and location parameters $\boldsymbol{\mu} = [-4 \;\; 0 \;\; 4]^\top$.

Covariance between two points

- The prior covariance between two points $\mathbf{x}_i$ and $\mathbf{x}_j$ is

  $$k\left(\mathbf{x}_i, \mathbf{x}_j\right) = \alpha\phi\left(\mathbf{x}_i\right)^\top \phi\left(\mathbf{x}_j\right),$$

  or in sum notation

  $$k\left(\mathbf{x}_i, \mathbf{x}_j\right) = \alpha \sum_{k=1}^{m} \phi_k\left(\mathbf{x}_i\right)\phi_k\left(\mathbf{x}_j\right)$$

- For the radial basis used this gives

  $$k\left(\mathbf{x}_i, \mathbf{x}_j\right) = \alpha \sum_{k=1}^{m} \exp\left(-\frac{\left|\mathbf{x}_i - \boldsymbol{\mu}_k\right|^2 + \left|\mathbf{x}_j - \boldsymbol{\mu}_k\right|^2}{2\ell^2}\right).$$

Covariance between two points

- The prior covariance between two points $\mathbf{x}_i$ and $\mathbf{x}_j$ is

  $$k\left(\mathbf{x}_i, \mathbf{x}_j\right) = \alpha\phi\left(\mathbf{x}_i\right)^\top \phi\left(\mathbf{x}_j\right),$$

  or in sum notation

  $$k\left(\mathbf{x}_i, \mathbf{x}_j\right) = \alpha\sum_{k=1}^{m} \phi_k\left(\mathbf{x}_i\right)\phi_k\left(\mathbf{x}_j\right)$$

- For the radial basis used this gives

  $$k\left(\mathbf{x}_i, \mathbf{x}_j\right) = \alpha\sum_{k=1}^{m} \exp\left(-\frac{\left|\mathbf{x}_i - \boldsymbol{\mu}_k\right|^2 + \left|\mathbf{x}_j - \boldsymbol{\mu}_k\right|^2}{2\ell^2}\right).$$

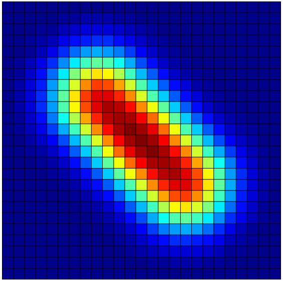
Covariance functions

RBF basis functions

$$k\left(\mathbf{x}, \mathbf{x}'\right) = \alpha \boldsymbol{\phi}(\mathbf{x})^\top \boldsymbol{\phi}(\mathbf{x}')$$

$$\phi_k(x) = \exp\left(-\frac{\|x - \mu_k\|_2^2}{\ell^2}\right)$$

$$\boldsymbol{\mu} = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$



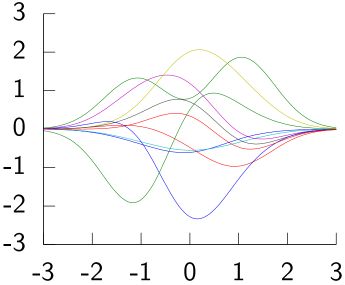
Covariance functions

RBF basis functions

$$k\left(\mathbf{x}, \mathbf{x}'\right) = \alpha \boldsymbol{\phi}(\mathbf{x})^\top \boldsymbol{\phi}(\mathbf{x}')$$

$$\phi_k(x) = \exp\left(-\frac{\|x-\mu_k\|_2^2}{\ell^2}\right)$$

$$\boldsymbol{\mu} = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$



Selecting number and location of basis

- Need to choose (1D input here):
  1. location of centers.
  2. number of basis functions.

- Consider uniform spacing over a region:

$$k\left(x_i, x_j\right) = \alpha \phi(x_i)^\top \phi(x_j)$$

Selecting number and location of basis

- Need to choose (1D input here):
  1. location of centers.
  2. number of basis functions.

- Consider uniform spacing over a region:

$$k\left(x_i, x_j\right) = \alpha \sum_{k=1}^{m} \phi_k(x_i)\phi_k(x_j)$$

Selecting number and location of basis

- Need to choose (1D input here):
  1. location of centers.
  2. number of basis functions.

- Consider uniform spacing over a region:

$$k\left(x_i, x_j\right) = \alpha \sum_{k=1}^{m} \exp\left(-\frac{\left(x_i - \mu_k\right)^2}{2\ell^2}\right) \exp\left(-\frac{\left(x_j - \mu_k\right)^2}{2\ell^2}\right)$$

Selecting number and location of basis

- Need to choose (1D input here):
  1. location of centers.
  2. number of basis functions.

- Consider uniform spacing over a region:

$$k\left(x_i, x_j\right) = \alpha \sum_{k=1}^{m} \exp\left(-\frac{(x_i - \mu_k)^2}{2\ell^2} - \frac{(x_j - \mu_k)^2}{2\ell^2}\right)$$

Selecting number and location of basis

- Need to choose (1D input here):
  1. location of centers.
  2. number of basis functions.

- Consider uniform spacing over a region:

$$k\left(x_i, x_j\right) = \alpha \sum_{k=1}^{m} \exp\left(-\frac{x_i^2 + x_j^2 - 2\mu_k\left(x_i + x_j\right) + 2\mu_k^2}{2\ell^2}\right).$$

Uniform basis functions

Set each center location to $\mu_k = a + \Delta\mu \cdot (k - 1)$ and specify the basis functions in terms of their indices.

$$
\begin{aligned}
k\left(x_i, x_j\right) =& \alpha' \Delta\mu \sum_{k=1}^{m} \exp\Bigg(- \frac{x_i^2 + x_j^2}{2\ell^2} \\
& - \frac{2\left(a + \Delta\mu \cdot (k-1)\right)\left(x_i + x_j\right) + 2\left(a + \Delta\mu \cdot (k-1)\right)^2}{2\ell^2}\Bigg).
\end{aligned}
$$

We've scaled variance of process by $\Delta\mu$.

Infinite basis functions

Take $\mu_1 = a$ and $\mu_m = b$ so $b = a + \Delta\mu \cdot (m-1)$, which implies $b - a = \Delta\mu(m-1)$ and therefore

$$m = \frac{b-a}{\Delta\mu} + 1$$

Take limit as $\Delta\mu \to 0$ so $m \to \infty$

$$k(x_i, x_j) = \alpha' \int_a^b \exp\left(-\frac{x_i^2 + x_j^2}{2\ell^2} + \frac{2\left(\mu - \frac{1}{2}\left(x_i + x_j\right)\right)^2 - \frac{1}{2}\left(x_i + x_j\right)^2}{2\ell^2}\right) \mathrm{d}\mu,$$

where we have used $a + k \cdot \Delta\mu \to \mu$.

Result

Performing the integration leads to

$$k(x_i,x_j) = \alpha' \sqrt{\pi \ell^2} \exp\left(-\frac{\left(x_i - x_j\right)^2}{4\ell^2}\right)$$

$$\times \frac{1}{2}\left[\operatorname{erf}\left(\frac{\left(b - \frac{1}{2}\left(x_i + x_j\right)\right)}{\ell}\right) - \operatorname{erf}\left(\frac{\left(a - \frac{1}{2}\left(x_i + x_j\right)\right)}{\ell}\right)\right],$$

Now take limit as $a \to -\infty$ and $b \to \infty$

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left(x_i - x_j\right)^2}{4\ell^2}\right).$$

where $\alpha = \alpha' \sqrt{\pi \ell^2}$.

Infinite feature space

- An RBF model with infinite basis functions is a Gaussian process.
- The covariance function is given by the exponentiated quadratic covariance function.

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left(x_i - x_j\right)^2}{4\ell^2}\right).$$

- Similar results can obtained for multi-dimensional input models.
- In many cases is easier to work with the kernel than with the basis.

1. **Probability helps to model any source of uncertainty.**
2. **The Gaussian appear in many operations in science.**
3. **Correlations are key in the samples from the Gaussian.**
4. **Gaussian processes are distributions over functions.**
5. **New covariances can be derived from old ones.**
6. **Prediction in GPs are linear algebra operations.**
7. **Marginal likelihood as objective for training. No CV.**
8. **Non Gaussian likelihoods require approximations.**
9. **An RBF model with infinite basis in as GP.**

Limitations of Gaussian processes

- Inference is $O(n^3)$ due to matrix inverse (in practice use Cholesky).
- Gaussian processes don't deal well with discontinuities (financial crises, phosphorylation, collisions, edges in images).
- Widely used exponentiated quadratic covariance (RBF) can be too smooth in practice (but there are many alternatives!).

Extensions

- Large scale GPs (large N). Nystron approximations, Sparse GPs, Fourier features.
- Unsupervised learning: We don't observe **X**. Gaussian Process Latent variable models.
- Multiple outputs/fidelities. Dealing simultaneously with several correlated outputs.
- **Deep Gaussian processes.** Convolution of stochastic processes. Useful to deal with non-stationary signals, discontinuities, etc.

Extensions

- Large scale GPs (large N). Nystron approximations, Sparse GPs, Fourier features.
- Unsupervised learning: We don't observe **X**. Gaussian Process Latent variable models.
- Multiple outputs/fidelities. Dealing simultaneously with several correlated outputs.
- **Deep Gaussian processes.** Convolution of stochastic processes. Useful to deal with non-stationary signals, discontinuities, etc.

Extensions

- Large scale GPs (large N). Nystron approximations, Sparse GPs, Fourier features.

- Unsupervised learning: We don't observe **X**. Gaussian Process Latent variable models.

- Multiple outputs/fidelities. Dealing simultaneously with several correlated outputs.

- **Deep Gaussian processes.** Convolution of stochastic processes. Useful to deal with non-stationary signals, discontinuities, etc.

Extensions

- Large scale GPs (large N). Nystron approximations, Sparse GPs, Fourier features.
- Unsupervised learning: We don't observe **X**. Gaussian Process Latent variable models.
- Multiple outputs/fidelities. Dealing simultaneously with several correlated outputs.
- **Deep Gaussian processes.** Convolution of stochastic processes. Useful to deal with non-stationary signals, discontinuities, etc.

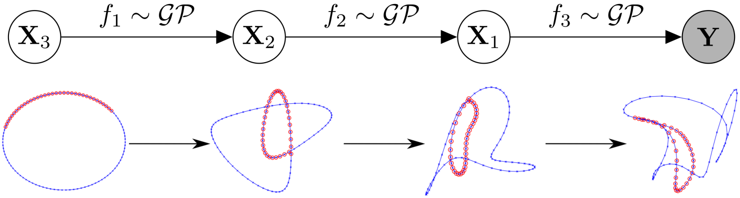
Deep Gaussian processes

Deep Gaussian processes

Damianou and Lawrence, [2013]

- Stochastic process resulting of composing several Gaussian process.
- Good for model non-stationary processes.
- The expressive power of a deep GP is significantly greater than that of a standard GP because the successive warping of latent variables.
- Active area of research.

Deep Gaussian processes



L layers of latent variables, $\{\mathbf{X}_l\}_{l=1}^{L}, \mathbf{X}_l \in \mathbb{R}^{N \times Q_l}$:

$$\mathbf{Y} = f_1(\mathbf{X}_1) + \epsilon_1, \quad \epsilon_1 \sim \mathcal{N}(0, \sigma_1^2 \mathbf{I})$$
$$\mathbf{X}_{l-1} = f_l(\mathbf{X}_l) + \epsilon_l, \quad \epsilon_l \sim \mathcal{N}(0, \sigma_l^2 \mathbf{I}), \quad l = 2 \ldots L$$

where $f_l(x) \sim \mathcal{GP}(0, k_l(x, x'))$.

Connections with neural networks

- **Reminder**: a GP is the limit of an infinitely wide RBF network.
- Deep GP: limit where the parametric function composition turns into a process composition.

Deep neural network:

$$\mathbf{g}(\mathbf{x}) = \mathbf{V}_L^\top \phi_L(\mathbf{W}_{L-1}\phi_{L-1}(\ldots \mathbf{W}_2\phi_1(\mathbf{U}_1\mathbf{x}))),$$

for $\mathbf{W}, \mathbf{U}, \mathbf{V}$ matrices of parameters and $\phi(\cdot)$ an activation.

Non-parametrically treating the stacked function composition $g(\mathbf{h}) = \mathbf{V}^\top \phi(\mathbf{U}\mathbf{h})$: deep GP

Connections with neural networks

- **Reminder**: a GP is the limit of an infinitely wide RBF network.
- Deep GP: limit where the parametric function composition turns into a process composition.

Deep neural network:

$$\mathbf{g}(\mathbf{x}) = \mathbf{V}_L^\top \phi_L(\mathbf{W}_{L-1}\phi_{L-1}(\ldots \mathbf{W}_2\phi_1(\mathbf{U}_1\mathbf{x}))),$$

for $\mathbf{W}, \mathbf{U}, \mathbf{V}$ matrices of parameters and $\phi(\cdot)$ an activation.

Non-parametrically treating the stacked function composition $g(\mathbf{h}) = \mathbf{V}^\top \phi(\mathbf{U}\mathbf{h})$: deep GP

Connections with neural networks

- **Reminder**: a GP is the limit of an infinitely wide RBF network.
- Deep GP: limit where the parametric function composition turns into a process composition.

Deep neural network:

$$\mathbf{g}(\mathbf{x}) = \mathbf{V}_L^\top \phi_L(\mathbf{W}_{L-1}\phi_{L-1}(\ldots \mathbf{W}_2\phi_1(\mathbf{U}_1\mathbf{x}))),$$

for $\mathbf{W}, \mathbf{U}, \mathbf{V}$ matrices of parameters and $\phi(\cdot)$ an activation.

Non-parametrically treating the stacked function composition $g(\mathbf{h}) = \mathbf{V}^\top \phi(\mathbf{U}\mathbf{h})$: deep GP

Inference in Deep GPs

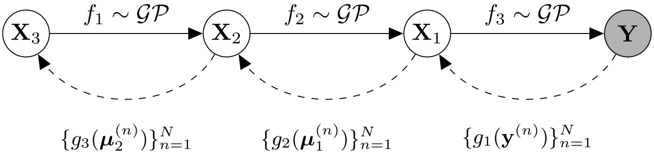
- In a standard GPs: inference by analytically integrating f.
- In the DGPs: all the latent variables have to additionally be integrated out:

$$p(\mathbf{Y}) = \int p(\mathbf{Y}|\mathbf{X}_1) \prod_{l=2}^{L} p(\mathbf{X}_{l-1}|\mathbf{X}_l) p(\mathbf{X}_L) \mathrm{d}\mathbf{X}_1 \ldots \mathrm{d}\mathbf{X}_L.$$

- Use approximated inference techniques: variational inference, Expectation Propagation (Damianou Lawrence, 2013; Bui et al., 2015), etc.
- Requires extra parameter per data point (we using a variational approximation).

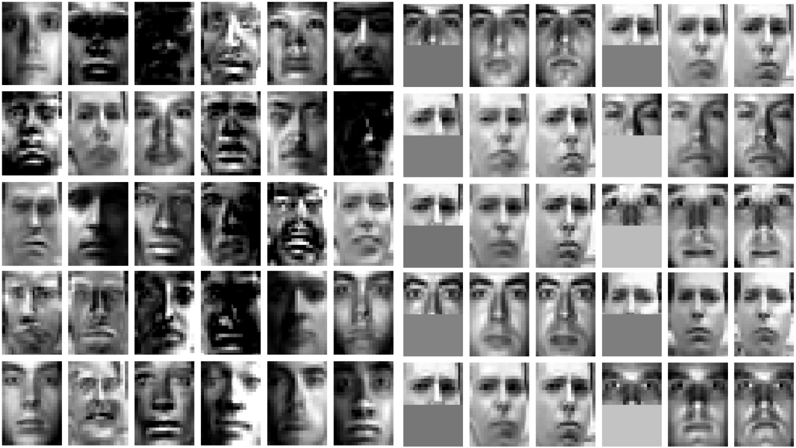
Extensions: Variationally auto-encoded deep GPs.

Dai, Damianou, Gonzalez, and Lawrence, [2016]



- Augmenting DGP with a variationally auto-encoded inference mechanism.
- Constrains the variational posterior distributions of latent variables.
- This allows us to reduce the number of parameters for optimization, which no longer grow linearly with the size of data.

Application: faces generation and imputation



1. **Probability helps to model any source of uncertainty.**

2. **The Gaussian appear in many operations in science.**

3. **Correlations are key in the samples from the Gaussian.**

4. **Gaussian processes are distributions over functions.**

5. **New covariances can be derived from old ones.**

6. **Prediction in GPs are linear algebra operations.**

7. **Marginal likelihood as objective for training. No CV.**

8. **Non Gaussian likelihoods require approximations.**

9. **An RBF model with infinite basis is a GP.**

10. **An non parametric treatment of NNs leads to a deep GP.**

GPSS: Gaussian Process Summer School

- `http://ml.dcs.shef.ac.uk/gpss/`
- Next one is in Sheffield in September 2018.

Many thanks to!!

Neil Lawrence, Zhenwen Dai, Andreas Damianou, Xiaoyu Lu, Mark Pullin, Andrei Paleyes, Maren Mahsereci, Vicky Scheider and Cliff McCollum.