CS-E4895 Gaussian Processes
Lecture 1: Introduction

Arno Solin

Aalto University

Monday 27.2.2023

Agenda for today

1. Course content, format, and evaluation
2. A very short intro to Gaussian Processes
3. Warm-up: Review of the multivariate Gaussian distribution

Course content

- *Gaussian processes (GPs) are a powerful machine learning paradigm for Bayesian nonparametric modelling. This course will give an overview of Gaussian processes in machine learning, and it provides both a theoretical and practical background for leveraging them. The course covers Gaussian process regression, classification, and unsupervised modelling, as well as a selection of more recent specialised topics.*

- We will cover
  - Gaussian process regression & classification
  - kernel learning
  - model selection
  - approximate inference & how to speed up GPs
  - spatio-temporal modelling
  - latent modelling
  - links to deep learning
  - GP theory

Format of the course

- The course will be based on
  - 12 lectures
  - 6 python notebook (jupyter) assignments
- To pass the course, you need to
  - Complete and hand in 6 weekly assignments for 5 ECTS
  - Attend exercise sessions
- This course has ran in various forms previously, but without its own course code. A lot of the material is based on previous runs, contributed by Michael Riis Andersen, William Wilkinson, Vincent Adam, Charles Gadd, Harri Lähdesmäki, and the current lecturers.

Course plan

Lectures

Time	Place	Lecturer	Topic
Mon, 27 Feb	U142	Arno Solin	1. Introduction & warm-up
Tue, 28 Feb	Y122	Ti John	2. Bayesian regression
Mon, 6 Mar	U142	Ti John	3. GP regression
Tue, 7 Mar	Y122	Aki Vehtari	4. Integration and model selection
Mon, 13 Mar	U142	Markus Heinonen	5. Kernel learning
Tue, 14 Mar	Y122	Arno Solin	6. GP classification
Mon, 20 Mar	U142	Arno Solin	7. Large-scale GPs
Tue, 21 Mar	Y122	Martin Trapp	8. GP theory
Mon, 27 Mar	U142	Markus Heinonen	9. Deep GPs
Tue, 28 Mar	Y122	Markus Heinonen	10. Latent modelling and unsupervised learning
Mon, 3 Apr	U142	Arno Solin	11. State-space GPs
Tue, 4 Apr	Y122	Aidan Scannell	12. Sequential decision-making

Lecturers

Arno Solin
1, 6–7, 11

Ti John
2–3

Aki Vehtari
4

Markus Heinonen
5, 9–10

Martin Trapp
8

Aidan Scannell
12

Teaching assistants

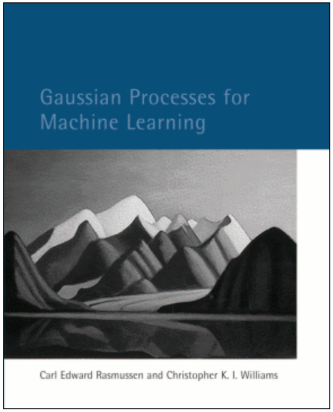
Severi Rissanen

Prakhar Verma

Main contact points for practical things on the course

Course material

- Lecture slides & Assignments
- The book *"Gaussian Processes for Machine Learning"* by Rasmussen and Williams, MIT press, 2006, `gaussianprocess.org/gpml` (available for download)



Assignments

Six jupyter notebook assignments

- Released on Wednesdays
- Complete at home and return following week by Thu exercise session
- Present solutions at exercise session (following week)
- First assignment released this week

Deadlines

- Weekly deadlines!

Grading

- Max 48 points: 6 points per assignment, 2 point per assignment by attending session
- Bonus: 2 bonus points for returning course feedback
- Grades: 1/5 24p, 2/5 28p, 3/5 32p, 4/5 36p, 5/5 40p

No exam

Relation to other courses

Target audience

- Designed as a 2nd / 1st year machine learning M.Sc. course

Prerequisites: Basics of ML

- CS-C3240 Machine Learning
- CS-E4710 Machine Learning: Supervised methods
- CS-E3210 Machine learning: Basic principles

Similar level courses

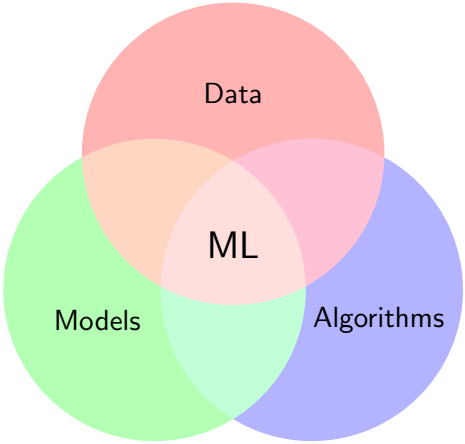
- CS-E5710 Bayesian Data Analysis (.. GPs are Bayesian)
- CS-E4820 Machine Learning: Advanced Probabilistic Methods (.. GPs are probabilistic)
- CS-E4830 Kernel Methods in Machine Learning (.. GPs are probabilistic kernel methods)
- CS-E4890 Deep Learning (.. GPs can do probabilistic deep learning)
- CS-E4800 Artificial Intelligence (.. GPs are often very practical for applied modelling)

Questions

A very short intro to Gaussian Processes

It's all about the tools you have in your toolbox

Gaussian processes and ML



Definitions

A random vector $\boldsymbol{x} = (x_1, x_2, \ldots, x_d)$ is said to have the **multivariate Gaussian distribution** if all linear combinations of $\boldsymbol{x}$ are Gaussian distributed:

$$y = a_1 x_1 + a_2 x_2 + \cdots + a_d x_d \sim \mathrm{N}(m, v)$$

for all $\boldsymbol{a} \in \mathbb{R}^d$

A **Gaussian process** (GP) is a collection of random variables over space, such that any finite subset of them have a joint Gaussian distribution.

Characterization and notation

- A Gaussian process can be considered as a distribution over functions $f : \mathcal{X} \to \mathbb{R}$ (the domain or index space $\mathcal{X}$ is typically $\mathbb{R}^d$)

$$f(\boldsymbol{x}) \sim \mathcal{GP}\big(\mu(\boldsymbol{x}), \kappa(\boldsymbol{x}, \boldsymbol{x}')\big)$$

- A Gaussian process is completely characterized by its mean function $\mu(\boldsymbol{x})$ and its covariance function $\kappa(\boldsymbol{x}, \boldsymbol{x}')$, which define

$$\mathbb{E}\left[f(\boldsymbol{x})\right] = \mu(\boldsymbol{x}) \quad \text{and} \quad \operatorname{cov}[f(\boldsymbol{x}), f(\boldsymbol{x}')] = \kappa(\boldsymbol{x}, \boldsymbol{x}')$$

Characterization and notation

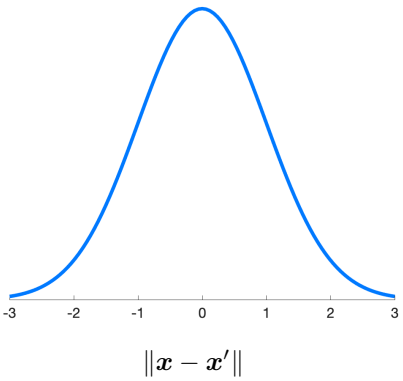
- The probability of any subset of function values $\boldsymbol{f} = f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_N)$ at any inputs $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ is
$$p(\boldsymbol{f}) = \mathrm{N}(\boldsymbol{f} \mid \boldsymbol{m}, \boldsymbol{K})$$
where $\boldsymbol{m} = \mu(\boldsymbol{x}_1), \ldots, \mu(\boldsymbol{x}_n)$ and $[\boldsymbol{K}]_{ij} = \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$
- If $\mathcal{X} = \mathbb{R}^d$, the GP prior describes infinitely many random variable $\left\{f(\boldsymbol{x}) : \boldsymbol{x} \in \mathbb{R}^d\right\}$, but in practice we only have to deal with a finite subset corresponding to the data set at hand, and where we want to evaluate ('test') the function
- This also gives rise to the *non-parametric* nature of GPs

Where the magic happens: The covariance function

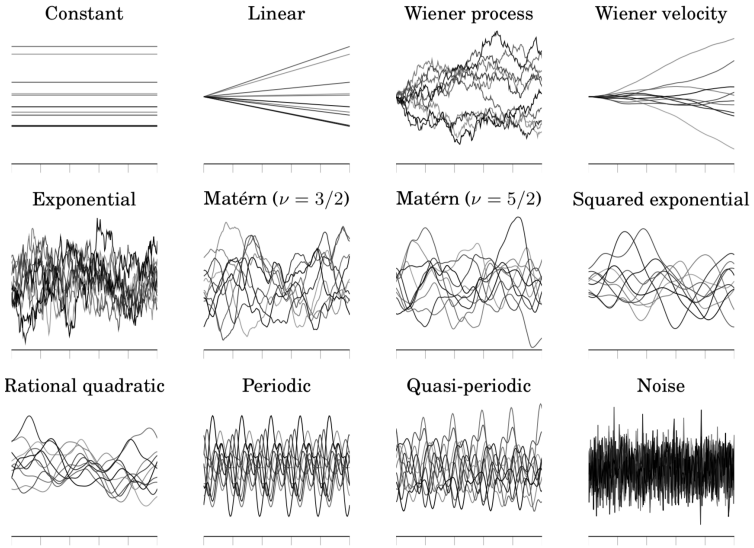
- In the kernel representation of GPs, the covariance function $\kappa(\boldsymbol{x}, \boldsymbol{x}')$ encodes prior beliefs of data-generating latent functions
- Typical choices are *continuity*, *differentiability* (smoothness), *periodicity*, *invariances*, *etc.*
- The RBF covariance function:

$$\kappa(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2 \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|^2}{2\ell^2} \right)$$

- The covariance functions typically have hyperparameters that are learned from data



Examples of draws from GP priors



Anatomy of a GP model in ML

In machine learning the kernel (moment) representation is favoured

$$f(\boldsymbol{x}) \sim \mathcal{GP}(\mu(\boldsymbol{x}), \kappa(\boldsymbol{x}, \boldsymbol{x}')) \qquad \textit{GP prior}$$

$$\boldsymbol{y} \mid \boldsymbol{f} \sim \prod_i p(y_i \mid f(\boldsymbol{x}_i)) \qquad \textit{likelihood}$$

Example: GP regression

- GP regression problem with input–output training pairs $\{(x_i, y_i)\}_{i=1}^n$:

$$f(x) \sim \mathrm{GP}(0, \kappa(x, x')),$$
$$y_i = f(x_i) + \varepsilon_i, \quad \varepsilon_i \sim \mathrm{N}(0, \sigma_\mathrm{n}^2)$$

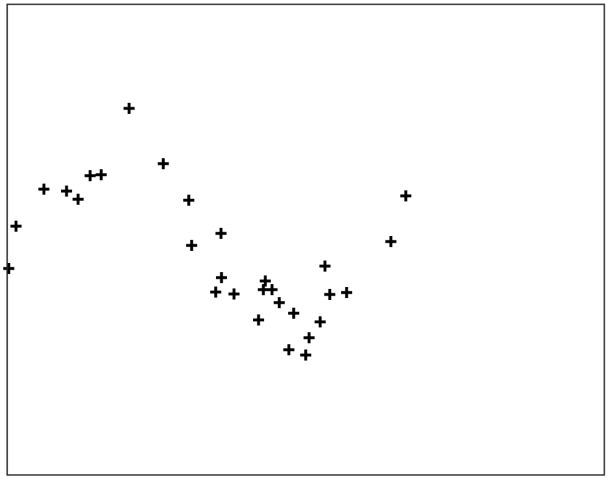
- The posterior mean and variance for an unseen test input x_* is given by:

$$\mathbb{E}[f_*] = \boldsymbol{k}_* \left(\boldsymbol{K} + \sigma_\mathrm{n}^2 \boldsymbol{I}\right)^{-1} \boldsymbol{y},$$
$$\mathbb{V}[f_*] = \kappa(x_*, x_*) - \boldsymbol{k}_* \left(\boldsymbol{K} + \sigma_\mathrm{n}^2 \boldsymbol{I}\right)^{-1} \boldsymbol{k}_*^\top$$

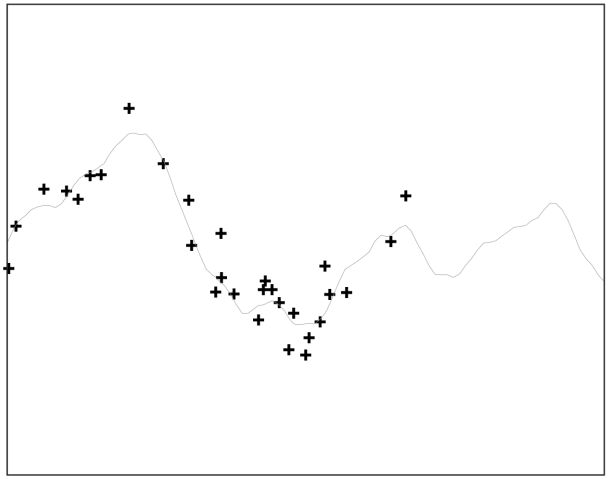
- Learn hyperparamters θ by maximizing w.r.t. log marginal likelihood:

$$\log p(\mathbf{y} \mid \theta) = -\frac{n}{2}\log(2\pi) - \frac{1}{2}\log|\boldsymbol{K}_\theta + \sigma_\mathrm{n}^2 \boldsymbol{I}| - \frac{1}{2}\boldsymbol{y}^\top \left(\boldsymbol{K}_\theta + \sigma_\mathrm{n}^2 \boldsymbol{I}\right)^{-1} \boldsymbol{y}$$

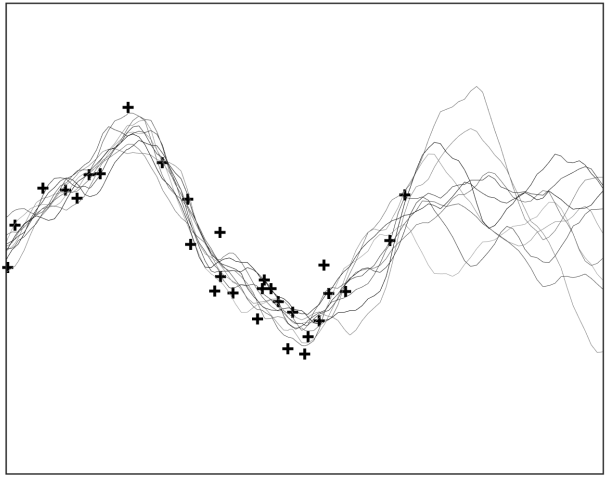
- Note the inversion of the $n \times n$ matrix.



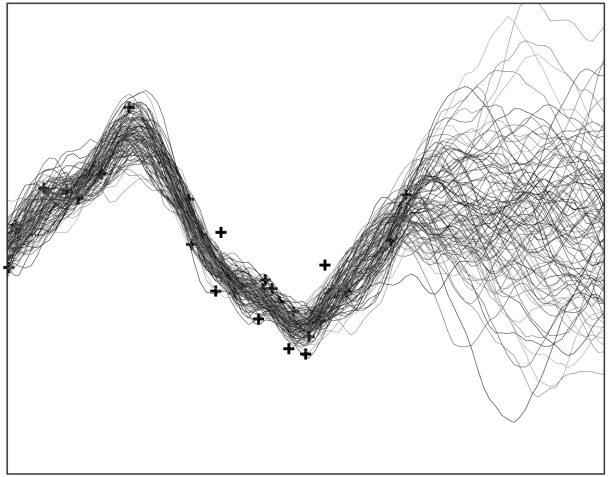
The input–output pairs



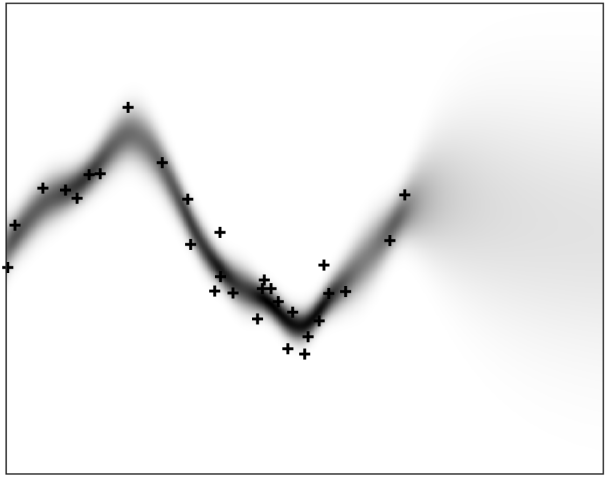
Draw from the GP posterior with a Matérn prior



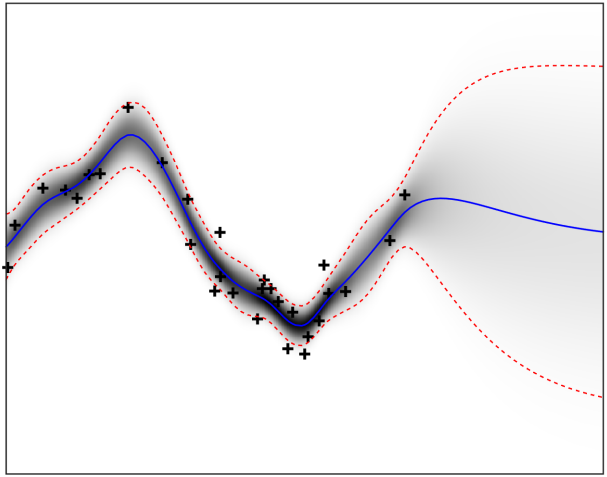
Draws from the GP posterior



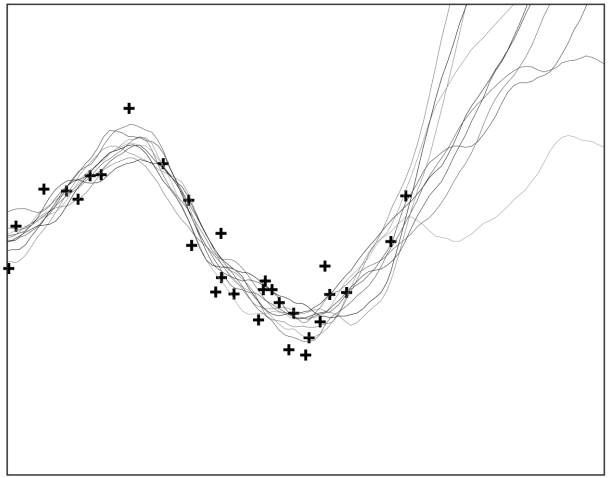
Draws from the GP posterior



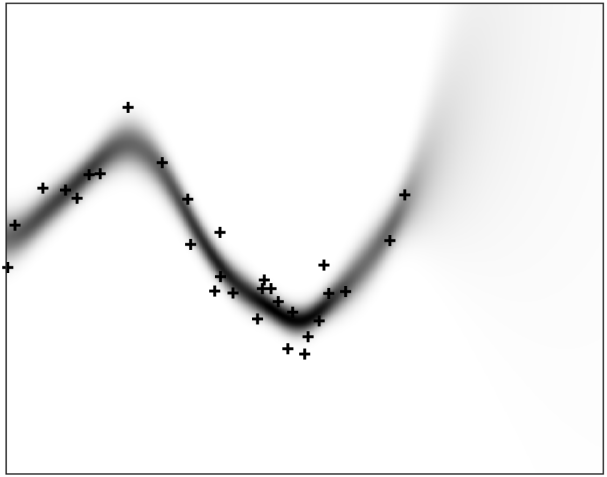
The GP posterior marginals



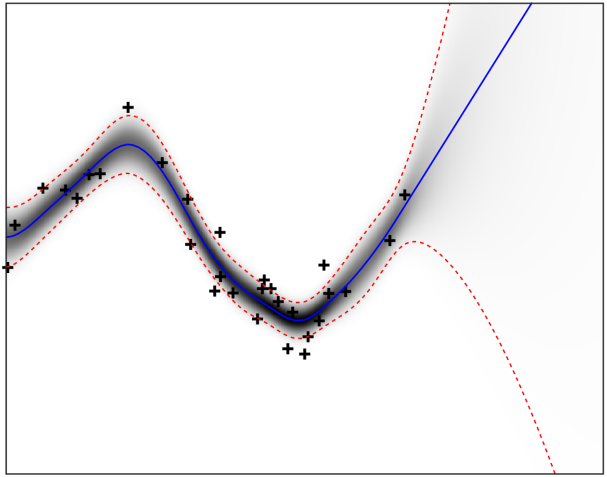
The stationary prior is mean-reverting



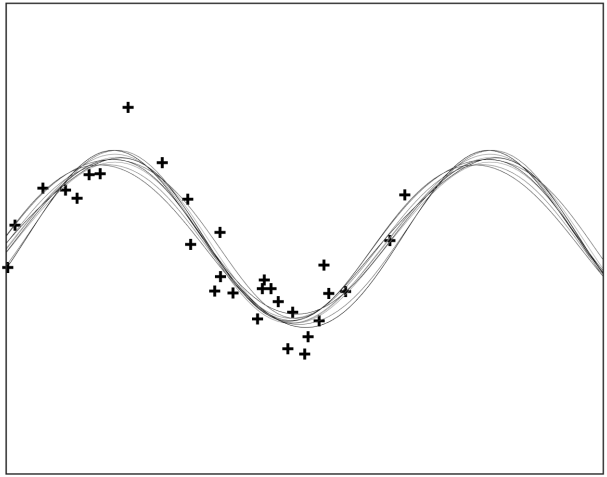
with a non-stationary prior



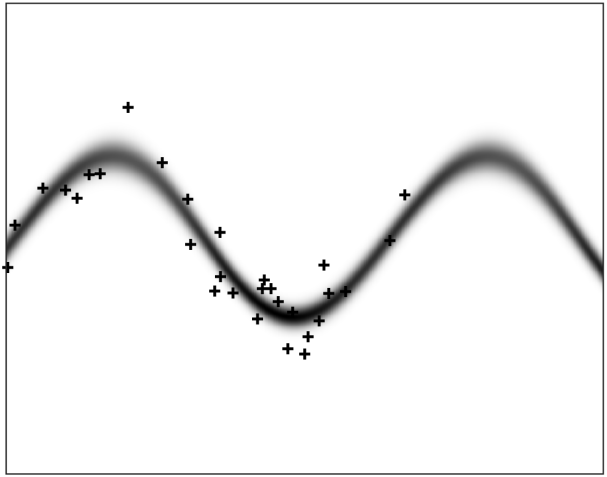
with a non-stationary prior



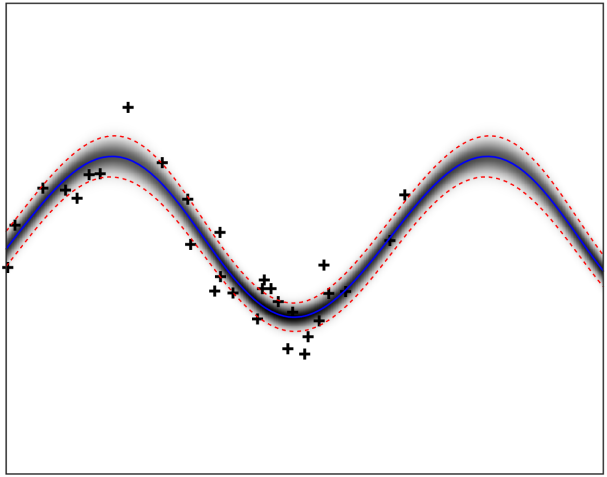
with a non-stationary prior



with a periodic prior



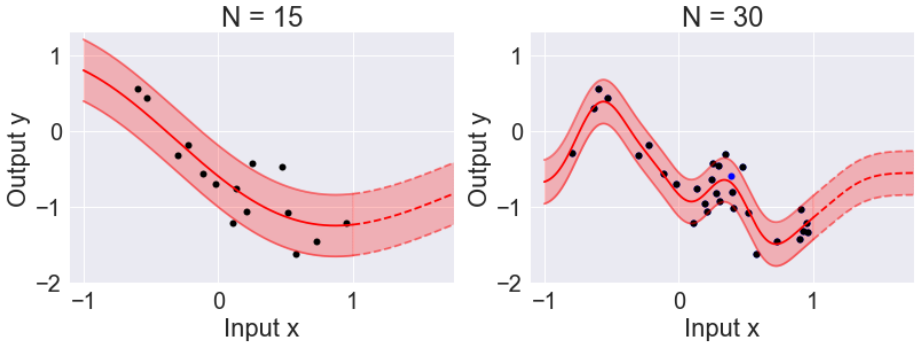
with a periodic prior



with a periodic prior

Gaussian processes in a nutshell

- It's all about learning functions from data
- Suppose we are given a data set $\mathcal{D} = \{\boldsymbol{x}_n, y_n\}_{n=1}^N$



- Gaussian processes (GPs) can
  - ... fit non-linear functions to data
  - ... make predictions for new inputs
  - ... provide sensible uncertainties
  - ... adjust model complexity to data (nonparametric)

Challenges that break the beauty

GPs have three challenges

Scaling to large data
A naïve solution to dealing with the expanded Gram (covariance) matrix requires $\mathcal{O}(n^3)$ compute and $\mathcal{O}(n^2)$ memory. Infeasible for $n > 10{,}000$.

Dealing with non-conjugate likelihoods
For a Gaussian observation model the GP posterior is available in closed-form. For non-conjugate likelihood models one has to resort to approximate inference methods.

Representational power
Gaussian processes are ideal for problems where it is easy to specify *meaningful* priors. For applications such as image classification this is hard.

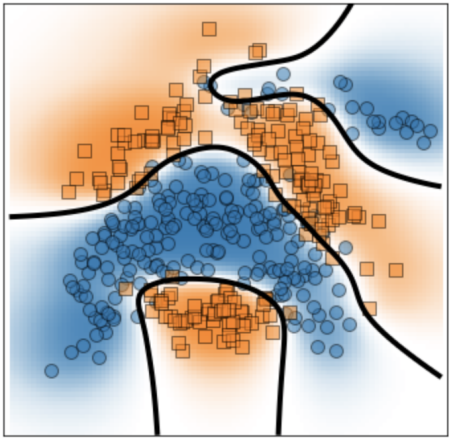
Scaling to large data

The naïve $\mathcal{O}(n^3)$ computational bottleneck ($\mathcal{O}(n^2)$ memory) can be tackled by

- Exploiting structure in the data
  (data on grid, inputs are in 1D, ...)
- Exploiting structure in the GP prior
  (GP prior is stationary, separable over input dimensions, ...)
- Solving the linear system approximately
  (conjugate-gradient solvers)
- Split problem into smaller chunks
  (local experts, subset of data, ...)
- Approximate the problem
  (Nyström, low-rank, inducing points, ...)
- Approximate the problem solution
  (SVGP = sparse (and stochastic) variational methods)

Dealing with non-conjugate likelihood models

- **MCMC (sampling) methods**
  (accurate but generally heavy)
- **Laplace approximation (LA)**
  (fast and simple)
- **Expectation propagation (EP)**
  (efficient but tricky)
- **Variational methods (VB/VI)**
  (popular but not problem-free)



GP classification with a Bernoulli likelihood

Representational power

- GPs can be seen as shallow, but infinitely wide models (see also deep GPs)
- Thus as such they are not ideal for problems where the data resides on some low-dimensional manifold in a high-dimensional space
- Instead, they can play a role as a building block of a larger model

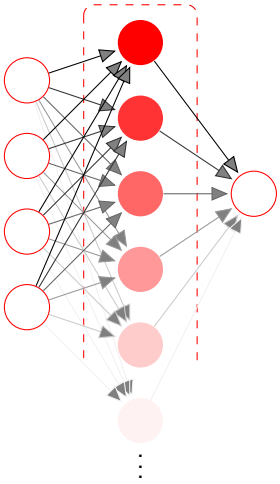
Connections and approaches to GPs

Connection to Neural Networks

- Radford Neal showed in the '90s that a random (untrained) single-layer feedforward network converges to a GP in the limit of infinite width.
- Let $\sigma(\cdot)$ be some non-linear (activation) function, and $\boldsymbol{w}$ and b be the network weights and biases.
- The associated kernel for the infinite-width network:

$$\kappa(\boldsymbol{x}, \boldsymbol{x}') = \int p(\boldsymbol{w})\, p(b)\, \sigma(\boldsymbol{w}^\top \boldsymbol{x} + b)\, \sigma(\boldsymbol{w}^\top \boldsymbol{x}' + b)\, \mathrm{d}\boldsymbol{w}\, \mathrm{d}b$$

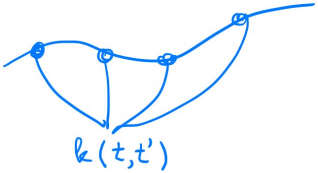
- The link can help analyze and understand NNs

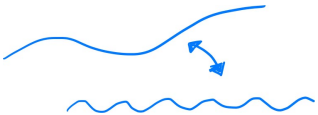


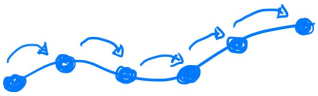
Connection to signal processing / SDEs

Alternative representations of GPs:

- Moment representation
  Considering the statistical properties of the input data jointly over time
- Spectral (Fourier) representation
  Analyzing the frequency-space representation of the problem/data
- State space (path) representation
  Description of sample behaviour as a dynamic system over time







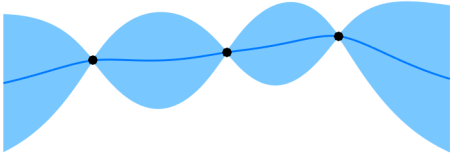
Connection to physics

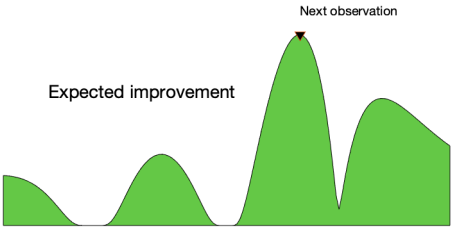
- First-principle models often written in terms of differential equations (ODEs, SDEs, PDEs, SPDEs)
- GPs used as structured priors ('latent forces') and for quantifying uncertainty
- GPs are preserved under linear operations (operating with linear operators)

Maxwell's equations induce a GP model for magnetic field variation

Connection to Bayesian optimization

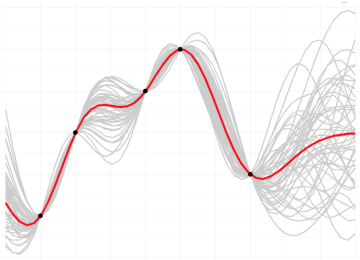
- Sometimes the objective function in an optimization problem is expensive to evaluate
- In Bayesian optimization, a GP prior is used for cleverly guide where to observe the objective function next

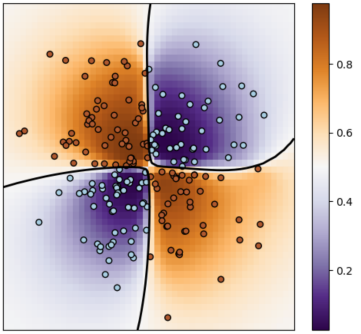




Multitude of Gaussian processes applications

- Regression (supervised learning)
  - Time series analysis / dynamical models
  - EEG brain imaging
  - Survival analysis for cancer data
  - Robot dynamics
  - Spatial modelling
- Classification (supervised learning)
  - Image recognition
  - Brain decoding
- Dimensionality reduction (unsupervised learning)
- Optimization of black box functions (Bayesian optimization)
- Numerical integration (Bayesian quadrature)
- Solving differential equations (probabilistic numerics)
- Experimental design / active learning
- Reinforcement learning





A very short intro to Gaussian Processes

- Gaussian processes provide a plug-and-play framework for probabilistic inference and learning
- Give an explicit way of injecting prior knowledge into a problem
- Provide meaningful uncertainty estimates and means for quantifying uncertainty

Properties of the multivariate Gaussian distribution

The multivariate Gaussian distribution

- **Definition** A random vector $\boldsymbol{x} = [x_1, x_2, \cdots, x_D]^\top$ is said to have the multivariate Gaussian distribution if all linear combinations of $\boldsymbol{x}$ are Gaussian distributed:

$$y = \boldsymbol{a}^\top \boldsymbol{x} = a_1 x_1 + a_2 x_2 + \cdots + a_D x_D \sim \mathcal{N}(m, v) \tag{1}$$

  for all $\boldsymbol{a} \in \mathbb{R}^D$, where $\boldsymbol{a} \neq \boldsymbol{0}$

- The multivariate Gaussian density for a variable $\boldsymbol{x} \in \mathbb{R}^D$:

$$\mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = (2\pi)^{-\frac{D}{2}} |\boldsymbol{\Sigma}|^{-\frac{1}{2}} \exp\left[-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})\right] \quad \in \mathbb{R}_{\geq 0} \tag{2}$$

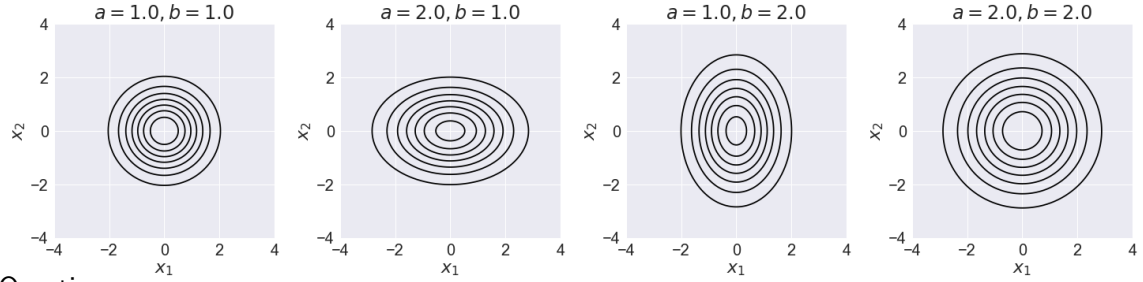
$$\log \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = -\frac{D}{2}\log 2\pi - \frac{1}{2}\log|\boldsymbol{\Sigma}| - \frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}) \quad \in \mathbb{R} \tag{3}$$

- Completely described by its parameters:
  - $\boldsymbol{\mu} \in \mathbb{R}^D$ is the mean vector
  - $\boldsymbol{\Sigma} \in \mathbb{R}^{D \times D}$ is the covariance matrix (positive definite)
- $(\boldsymbol{\Sigma})_{ij}$ is the covariance between the i'th and j'th elements x_i and x_j of $\boldsymbol{x}$

Interpretation of the covariance matrix - 2D examples

The diagonal of the covariance controls the scaling/marginal variances

$$\boldsymbol{\mu} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad \boldsymbol{\Sigma} = \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix} \tag{4}$$



Questions:

1. If $\boldsymbol{\Sigma}$ is diagonal, then x_1 and x_2 are uncorrelated? True or false?
2. If $\boldsymbol{\Sigma}$ is diagonal, then x_1 and x_2 are independent? True or false?
3. What is the volume (integral) of the density?
4. Which of the four densities has the highest peak and why?

The density at the mode

- The density is given by

$$\mathcal{N}\left(\boldsymbol{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}\right) = (2\pi)^{-\frac{D}{2}} \left|\boldsymbol{\Sigma}\right|^{-\frac{1}{2}} \exp\left[-\frac{1}{2}\left(\boldsymbol{x}-\boldsymbol{\mu}\right)^{\top} \boldsymbol{\Sigma}^{-1}\left(\boldsymbol{x}-\boldsymbol{\mu}\right)\right] \tag{5}$$

- The mode (highest density value) is achieve at $\boldsymbol{x} = \boldsymbol{\mu}$

$$\mathcal{N}\left(\boldsymbol{\mu} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}\right) = (2\pi)^{-\frac{D}{2}} \left|\boldsymbol{\Sigma}\right|^{-\frac{1}{2}} \tag{6}$$

- The determinant of the covariance is

$$\left|\boldsymbol{\Sigma}\right| = \mathbf{det}\begin{bmatrix} a & \rho \\ \rho & b \end{bmatrix} = ab - \rho^2 \tag{7}$$

- Therefore

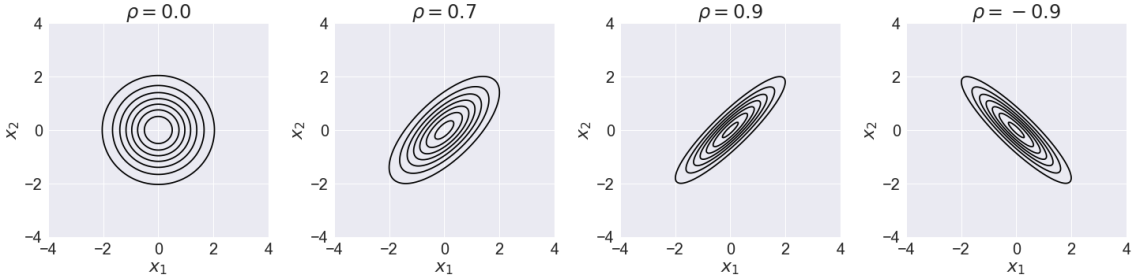
$$\mathcal{N}\left(\boldsymbol{\mu} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}\right) = (2\pi)^{-\frac{D}{2}} \left|\boldsymbol{\Sigma}\right|^{-\frac{1}{2}} = \frac{(2\pi)^{-\frac{D}{2}}}{\sqrt{ab-\rho^2}} \tag{8}$$

Interpretation of the covariance matrix

The off-diagonals control the covariances:

$$(\mathbf{\Sigma})_{ij} = \mathbf{cov}\,(x_i, x_j) = \mathbb{E}\,[x_i x_j] - \mu_i \mu_j \tag{9}$$

$$\boldsymbol{\mu} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad \mathbf{\Sigma} = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix} \tag{10}$$



Question:

- Which of the four densities has the highest peak and why?

Interpretation of the covariance matrix

Covariance matrices must be symmetric:

$$(\mathbf{\Sigma})_{ij} = \text{cov}\,(x_i, x_j) = \text{cov}\,(x_j, x_i) = (\mathbf{\Sigma})_{ji} \tag{11}$$

Consider the following set of covariance matrices:

$$\mathbf{\Sigma} = \begin{bmatrix} a & c \\ c & b \end{bmatrix} \tag{12}$$

c is the covariance between x_1 and x_2. Can c take any values?

$$\det\mathbf{\Sigma} = ab - c^2 \geq 0 \qquad \Rightarrow \qquad \left|c\right| \leq \sqrt{a}\sqrt{b} \tag{13}$$

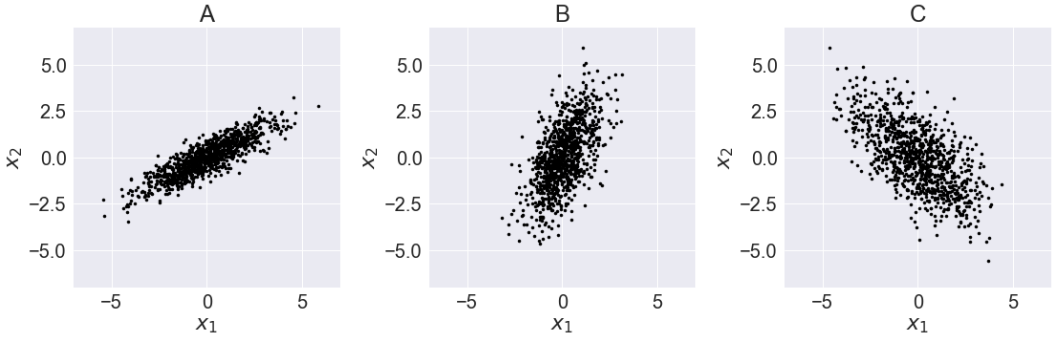
$\mathbf{\Sigma}$ must be positive definite

Interpretation of the covariance matrix

Determine which of the following 5 matrices are valid covariance matrices and match them to the set of samples below.

$$\boldsymbol{\Sigma}_1 = \begin{bmatrix} 3 & -2 \\ -2 & 3 \end{bmatrix} \qquad \boldsymbol{\Sigma}_2 = \begin{bmatrix} 3 & 2 \\ 1.5 & 3 \end{bmatrix} \qquad \boldsymbol{\Sigma}_3 = \begin{bmatrix} 1 & 1 \\ 1 & 3 \end{bmatrix}$$

$$\boldsymbol{\Sigma}_4 = \begin{bmatrix} 1 & -2 \\ -2 & 3 \end{bmatrix} \qquad \boldsymbol{\Sigma}_5 = \begin{bmatrix} 3 & 1.5 \\ 1.5 & 1 \end{bmatrix}$$



The multivariate Gaussian: Basic properties

- Gaussian distributions are closed under addition:

$$\boldsymbol{x}_1 \sim \mathcal{N}(\boldsymbol{m}_1, \boldsymbol{V}_1), \quad \boldsymbol{x}_2 \sim \mathcal{N}(\boldsymbol{m}_2, \boldsymbol{V}_2) \quad \Rightarrow \quad \boldsymbol{x}_1 + \boldsymbol{x}_2 \sim \mathcal{N}(\boldsymbol{m}_1 + \boldsymbol{m}_2, \boldsymbol{V}_1 + \boldsymbol{V}_2) \tag{14}$$

- For any finite number of independent variables:

$$\boldsymbol{x}_i \sim \mathcal{N}(\boldsymbol{m}_i, \boldsymbol{V}_i) \quad \Rightarrow \quad \sum_i \boldsymbol{x}_i \sim \mathcal{N}\left(\sum_i \boldsymbol{m}_i, \sum_i \boldsymbol{V}_i\right) \tag{15}$$

- Gaussian distributions are closed under affine transformations:

$$\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{m}, \boldsymbol{V}), \quad \Rightarrow \quad \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b} \sim \mathcal{N}\left(\boldsymbol{A}\boldsymbol{m} + \boldsymbol{b}, \boldsymbol{A}\boldsymbol{V}\boldsymbol{A}^\top\right) \tag{16}$$

- Manipulating Gaussian distributions often boils down to linear algebra
- The 'Matrix cookbook' (Section 8) and Rasmussen's book (Appendix A)

Question

... how to use the following two results

$$\boldsymbol{x}_i \sim \mathcal{N}\left(\boldsymbol{m}_i, \boldsymbol{V}_i\right) \quad \Rightarrow \quad \sum_i \boldsymbol{x}_i \sim \mathcal{N}\left(\sum_i \boldsymbol{m}_i, \sum_i \boldsymbol{V}_i\right) \tag{17}$$

$$\boldsymbol{x} \sim \mathcal{N}\left(\boldsymbol{m}, \boldsymbol{V}\right) \quad \Rightarrow \quad \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b} \sim \mathcal{N}\left(\boldsymbol{A}\boldsymbol{m} + \boldsymbol{b}, \boldsymbol{A}\boldsymbol{V}\boldsymbol{A}^\top\right), \tag{18}$$

to calculate the distribution of $\boldsymbol{Y}$ in the following linear model?

$$\boldsymbol{Y} = \boldsymbol{\mu} + \boldsymbol{X}\boldsymbol{w} + \boldsymbol{\epsilon}, \tag{19}$$

where

$$\boldsymbol{w} \sim \mathcal{N}\left(\boldsymbol{m}, \boldsymbol{V}\right) \qquad \boldsymbol{\epsilon} \sim \mathcal{N}\left(\boldsymbol{0}, \sigma^2 \boldsymbol{I}\right) \tag{20}$$

Sampling from the multivariate Gaussian distribution

$$\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{m}, \boldsymbol{V}) \quad \Rightarrow \quad \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b} \sim \mathcal{N}\left(\boldsymbol{A}\boldsymbol{m} + \boldsymbol{b}, \boldsymbol{A}\boldsymbol{V}\boldsymbol{A}^\top\right) \tag{21}$$

- Suppose we know how to generate samples from a standardized univariate Gaussian distribution
- How can we use the above result to generate samples from an arbitrary multivariate Gaussian distribution $\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{m}, \boldsymbol{V})$?
  1. Compute the matrix square root of $\boldsymbol{V} = \boldsymbol{L}\boldsymbol{L}^\top$
  2. Generate a sample of $\boldsymbol{x}$ such that $x_i \sim \mathcal{N}(0, 1)$, i.e. $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$
  3. Compute $\boldsymbol{y} = \boldsymbol{L}\boldsymbol{x} + \boldsymbol{m}$
- Why does it work?

$$\boldsymbol{y} = \boldsymbol{L}\boldsymbol{x} + \boldsymbol{m} \sim \mathcal{N}\left(\boldsymbol{L}\boldsymbol{0} + \boldsymbol{m}, \boldsymbol{L}\boldsymbol{I}\boldsymbol{L}^\top\right) = \mathcal{N}(\boldsymbol{m}, \boldsymbol{V}) \tag{22}$$

The multivariate Gaussian: Marginalization

- Gaussian densities are closed under marginalization
- Let $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ be a partitioning of $\boldsymbol{x} = \boldsymbol{x}_1 \cup \boldsymbol{x}_2$, then

$$p(\boldsymbol{x}_1, \boldsymbol{x}_2) = \mathcal{N}\left(\begin{bmatrix}\boldsymbol{x}_1 \\ \boldsymbol{x}_2\end{bmatrix} \mid \begin{bmatrix}\boldsymbol{m}_1 \\ \boldsymbol{m}_2\end{bmatrix}, \begin{bmatrix}\boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22}\end{bmatrix}\right) \tag{23}$$

then

$$p(\boldsymbol{x}_1) = \int p(\boldsymbol{x}_1, \boldsymbol{x}_2)\, \mathrm{d}\boldsymbol{x}_2 = \mathcal{N}\left(\boldsymbol{x}_1 \mid \boldsymbol{m}_1, \boldsymbol{\Sigma}_{11}\right) \tag{24}$$

and

$$p(\boldsymbol{x}_2) = \int p(\boldsymbol{x}_1, \boldsymbol{x}_2)\, \mathrm{d}\boldsymbol{x}_1 = \mathcal{N}\left(\boldsymbol{x}_2 \mid \boldsymbol{m}_2, \boldsymbol{\Sigma}_{22}\right) \tag{25}$$

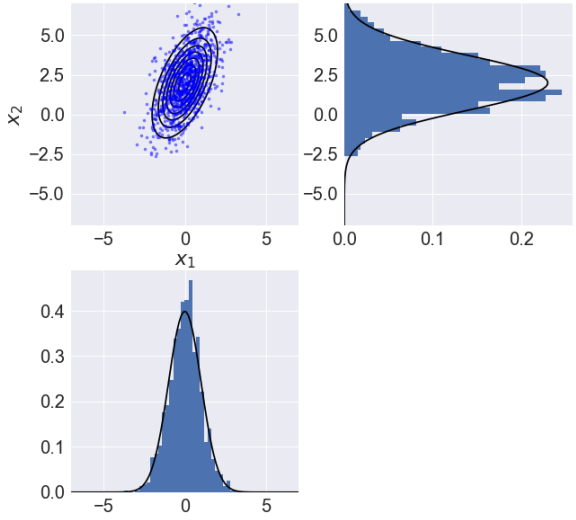
- The same is true for any partitioning

Marginalization example in 2D

$$\boldsymbol{x} \sim \mathcal{N}\left(\begin{bmatrix}0\\2\end{bmatrix}, \begin{bmatrix}1 & 1\\1 & 3\end{bmatrix}\right)$$

$$x_1 \sim \mathcal{N}(0, 1)$$

$$x_2 \sim \mathcal{N}(2, 3)$$



Conditioning

- Gaussian densities are closed under conditioning!
- Recall the definition of conditioning:

$$p(A \mid B) = \frac{p(A \cap B)}{p(B)} \tag{26}$$

- Let $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ be a partitioning of $\boldsymbol{x} = \boldsymbol{x}_1 \cup \boldsymbol{x}_2$, then

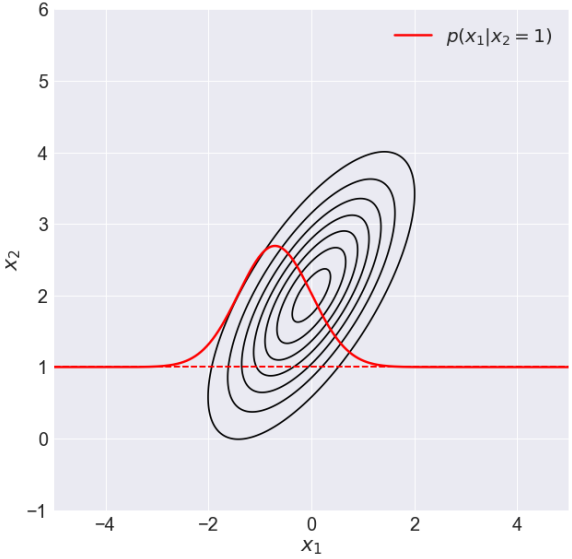
$$p(\boldsymbol{x}_1, \boldsymbol{x}_2) = \mathcal{N}\left(\begin{bmatrix}\boldsymbol{x}_1 \\ \boldsymbol{x}_2\end{bmatrix} \mid \begin{bmatrix}\boldsymbol{m}_1 \\ \boldsymbol{m}_2\end{bmatrix}, \begin{bmatrix}\boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22}\end{bmatrix}\right) \tag{27}$$

- The conditional of x_1 is given x_2 by:

$$p(\boldsymbol{x}_1 \mid \boldsymbol{x}_2) = \mathcal{N}\left(\boldsymbol{x}_1 \mid \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\left[\boldsymbol{x}_2 - \boldsymbol{m}_2\right] + \boldsymbol{m}_1, \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21}\right) \tag{28}$$

  - $\boldsymbol{x}_1$ is a random variable, $\boldsymbol{x}_2$ is assigned a fixed value

Conditioning example in 2D



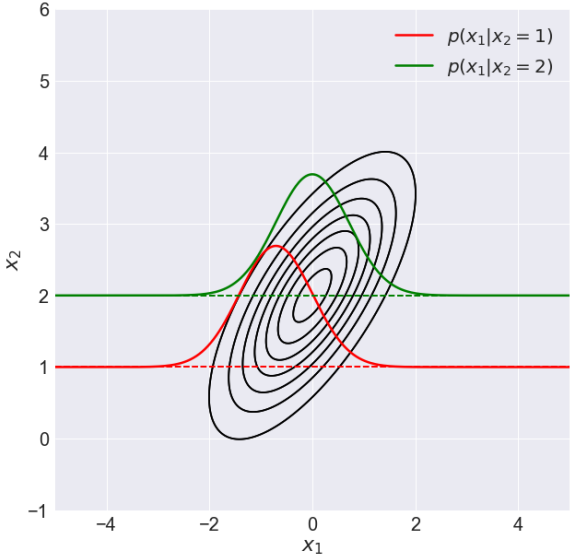
- 2D example

$$\boldsymbol{\mu} = \begin{bmatrix} 0 \\ 2 \end{bmatrix} \qquad \boldsymbol{\Sigma} = \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix}$$

- Assume we observe $x_2 = 1$
- The conditional disitribution

$$p(x_1 \mid x_2) = \mathcal{N}\left(x_1 | -\frac{\sqrt{2}}{2}, \frac{1}{2}\right)$$

Conditioning example in 2D



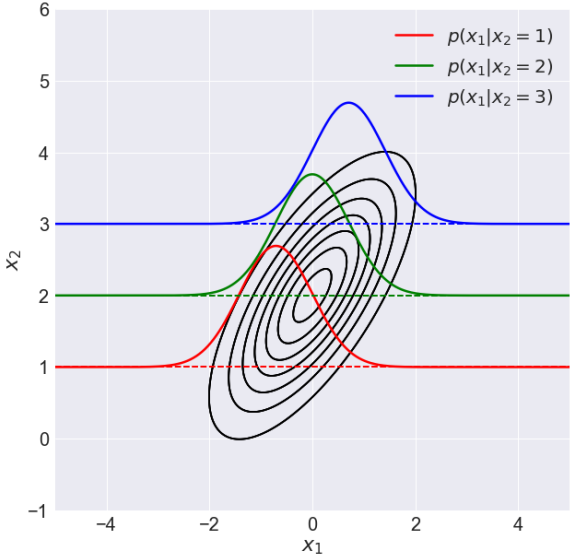
- 2D example

$$\boldsymbol{\mu} = \begin{bmatrix} 0 \\ 2 \end{bmatrix} \qquad \boldsymbol{\Sigma} = \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix}$$

- Assume we observe $x_2 = 2$
- The conditional disitribution

$$p(x_1 \mid x_2) = \mathcal{N}\left(x_1 | 0, \frac{1}{2}\right)$$

Conditioning example in 2D



- 2D example

$$\boldsymbol{\mu} = \begin{bmatrix} 0 \\ 2 \end{bmatrix} \qquad \boldsymbol{\Sigma} = \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix}$$

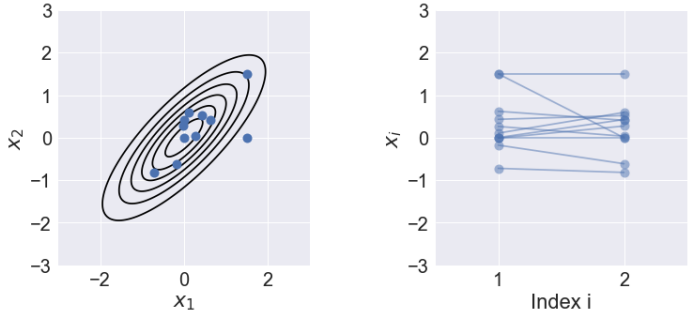
- Assume we observe $x_2 = 3$
- The conditional disitribution

$$p(x_1 \mid x_2) = \mathcal{N}\left(x_1 | \frac{\sqrt{2}}{2}, \frac{1}{2}\right)$$

Visualizing samples in higher dimensions

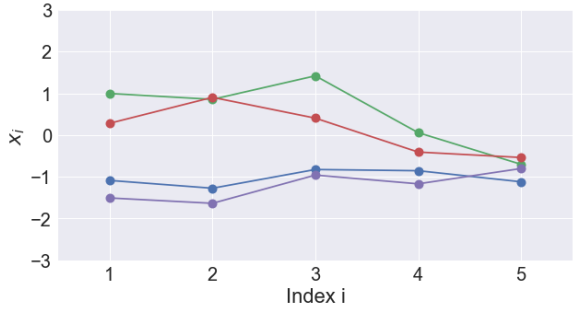
- Visualizations in 2D

$$\mathbf{\Sigma} = \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix}$$



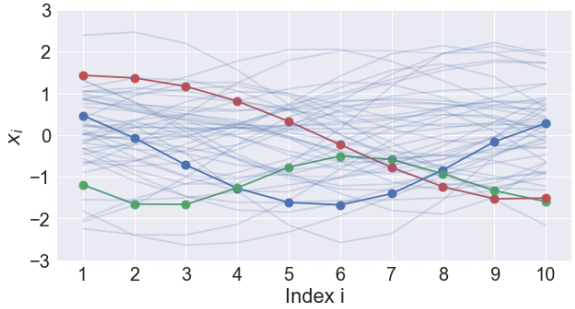
Visualizing samples in higher dimensions

- Visualizations in 5D



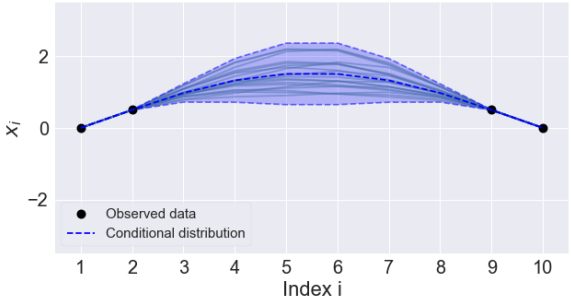
Visualizing samples in higher dimensions

- Visualizations in 10D



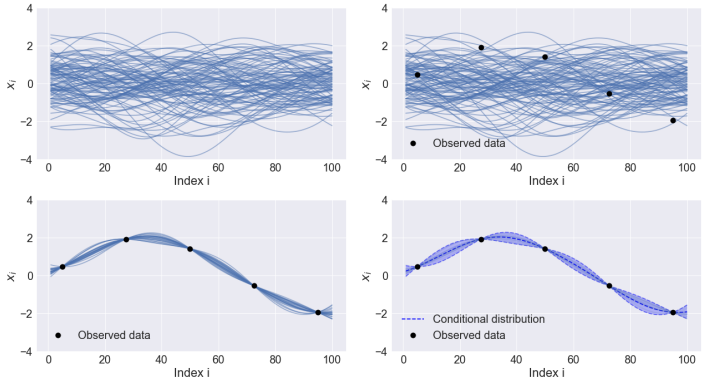
Back to conditioning

- So far, we have seen samples from the distribution $p(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x} \mid \mathbf{0}, \boldsymbol{\Sigma})$
- We can also write $p(\boldsymbol{x}) = p(x_1, \boldsymbol{x}_{2:10})$
- We now observe $x_1 = 0$
- Let's sample from the conditional distribution $p(\boldsymbol{x}_{2:10} \mid x_1 = 0)$



Back to conditioning II

- Let's now consider a case with $\boldsymbol{x} \in \mathbb{R}^{100}$ dimensions with 5 observations



- Informally: We can think of functions as vectors with infinite dimensions
- Using conditining in Gaussian distributions, we can do non-linear regression!

The end of today's lecture

- Tomorrow on Tuesday at 10 am
  - Ti will introduce Gaussian processes more formally
  - Read Chapter 1 & 2 of the Gaussian process book `gaussianprocess.org/gpml`