* * * Artificial Intelligence and Machine Learning * * *

# Advanced Regression Techniques

Daniel Wehner, M.Sc.

SAP SE / DHBW Mannheim

Summer term 2025



Find all slides on GitHub (DaWe1992/Applied_ML_Fundamentals)

# Lecture Overview

**I** Machine Learning Introduction
**II** Optimization Techniques
**III** Bayesian Decision Theory
**IV** Non-parametric Density Estimation
**V** Probabilistic Graphical Models
**VI** Linear Regression
**VII** Logistic Regression
**VIII** Deep Learning
**IX** Evaluation
**X** Decision Trees
**XI** Support Vector Machines
**XII** Clustering
**XIII** Principal Component Analysis
**XIV** Reinforcement Learning
• **XV** Advanced Regression

# Agenda for this Unit

1. Operations on GAUSSian Distributions
2. BAYESian Linear Regression
3. Kernel Ridge Regression
4. GAUSSian Process Regression
5. Wrap-Up

Section:

# Operations on Gaussian Distributions


Gaussian Distribution
Conditional Gaussian Distributions
Marginal Gaussian Distributions
Bayes' Theorem for Gaussian Variables


# Univariate GAUSSian Distribution

**Univariate GAUSSian distribution:**

$$\mathcal{N}\left(x;\mu,\sigma^2\right) := \frac{1}{\sqrt{2\pi\sigma^2}} \cdot \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{1}$$

- A univariate GAUSSian random variable is uniquely identified by its **mean** $\mu \in \mathbb{R}$ and **variance** $\sigma^2 \in \mathbb{R}^+ := \left\{x \in \mathbb{R} : x \geqslant 0\right\}$
- GAUSSian distributions have nice **analytic properties** which we shall exploit

# Multivariate Gaussian Distribution

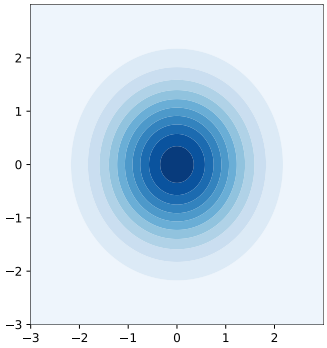
**Multivariate Gaussian distribution:**

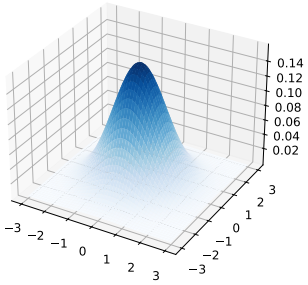
$$\mathcal{N}_D(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) := \frac{1}{\sqrt{(2\pi)^D \det(\boldsymbol{\Sigma})}} \cdot \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right) \quad (2)$$

- $D \in \mathbb{N}$ is the dimension
- For $D = 1$ definition (2) reduces to the univariate Gaussian distribution (1)
- A multidimensional Gaussian random variable is uniquely identified by its **mean** $\boldsymbol{\mu} \in \mathbb{R}^D$ and its **covariance matrix** $\boldsymbol{\Sigma} \in \mathbb{R}^{D \times D}$

# Visualization of a multivariate Gaussian Distribution

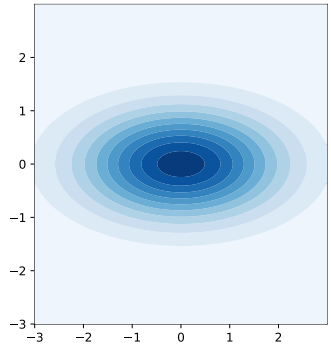
$$\boldsymbol{\Sigma} = \begin{pmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{pmatrix}$$

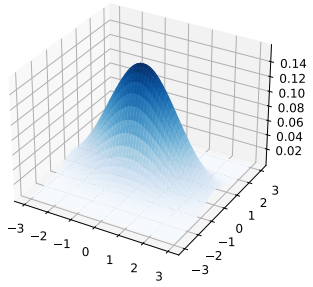


# Visualization of a multivariate Gaussian Distribution (Ctd.)

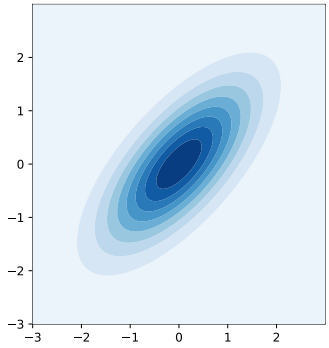
$$\boldsymbol{\Sigma} = \begin{pmatrix} 2.0 & 0.0 \\ 0.0 & 0.5 \end{pmatrix}$$

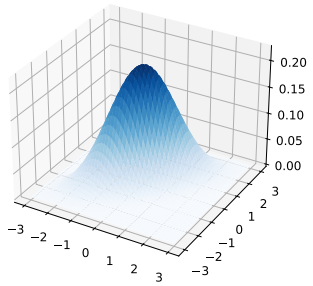


# Visualization of a multivariate Gaussian Distribution (Ctd.)

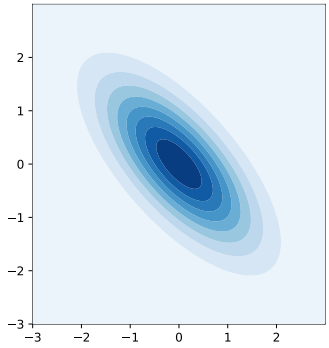
$$\boldsymbol{\Sigma} = \begin{pmatrix} 1.0 & 0.7 \\ 0.7 & 1.0 \end{pmatrix}$$

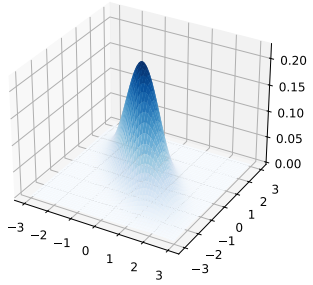


# Visualization of a multivariate Gaussian Distribution (Ctd.)

$$\boldsymbol{\Sigma} = \begin{pmatrix} 1.0 & -0.7 \\ -0.7 & 1.0 \end{pmatrix}$$



# Partitioned Gaussian Distribution

- In the following we introduce **marginalization** and **conditioning** of multivariate Gaussian random variables
- Let $\boldsymbol{x} \in \mathbb{R}^D$ be a Gaussian random variable which we partition into two disjoint sets $\boldsymbol{x}_a$ and $\boldsymbol{x}_b$:

$$p(\boldsymbol{x}) = p\left(\begin{bmatrix}\boldsymbol{x}_a \\ \boldsymbol{x}_b\end{bmatrix}\right) = \mathcal{N}\left(\begin{bmatrix}\boldsymbol{\mu}_a \\ \boldsymbol{\mu}_b\end{bmatrix}, \begin{bmatrix}\boldsymbol{\Sigma}_{aa} & \boldsymbol{\Sigma}_{ab} \\ \boldsymbol{\Sigma}_{ba} & \boldsymbol{\Sigma}_{bb}\end{bmatrix}\right) \tag{3}$$

- The variables $\boldsymbol{x}_a \in \mathbb{R}^K$ and $\boldsymbol{x}_b \in \mathbb{R}^{D-K}$ are **jointly Gaussian**
- $\boldsymbol{\Sigma}_{aa}$ and $\boldsymbol{\Sigma}_{bb}$ are symmetric matrices and we have $\boldsymbol{\Sigma}_{ba} = \boldsymbol{\Sigma}_{ab}^\top$


# Conditional GAUSSian Distribution

- Suppose we observe the variables $\boldsymbol{x}_b$ and want to compute the **conditional probability** $p(\boldsymbol{x}_a|\boldsymbol{x}_b)$
- This probability is distributed GAUSSian with mean $\boldsymbol{\mu}_{a|b}$ and covariance $\boldsymbol{\Sigma}_{a|b}$:

$$p(\boldsymbol{x}_a|\boldsymbol{x}_b) = \mathcal{N}\big(\boldsymbol{x}_a; \boldsymbol{\mu}_{a|b}, \boldsymbol{\Sigma}_{a|b}\big) \tag{4}$$

- The parameters $\boldsymbol{\mu}_{a|b}$ and $\boldsymbol{\Sigma}_{a|b}$ can be computed in closed form:

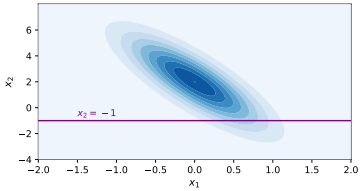
$$\boldsymbol{\mu}_{a|b} := \boldsymbol{\mu}_a + \boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}(\boldsymbol{x}_b - \boldsymbol{\mu}_b) \tag{5}$$

$$\boldsymbol{\Sigma}_{a|b} := \boldsymbol{\Sigma}_{aa} - \boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}\boldsymbol{\Sigma}_{ba} \tag{6}$$

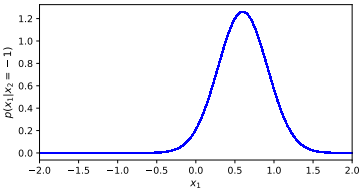
# Example: Conditional Gaussian Distribution

- Consider the Gaussian

$$p(x_1, x_2) = \mathcal{N}\left(\begin{bmatrix}0\\2\end{bmatrix}, \begin{bmatrix}0.3 & -1\\-1 & 5\end{bmatrix}\right)$$




- **Question:** What is $p(x_1|x_2 = -1)$?
- According to equations (5) and (6) we obtain $\mu_{x_1|x_2} = 0.6$ and $\sigma^2_{x_1|x_2} = 0.1$
- Thus, $p(x_1|x_2 = -1) = \mathcal{N}(0.6, 0.1)$




# Marginal GAUSSian Distribution

- Sometimes we may be interested in the **marginal distribution** $p(\boldsymbol{x}_a)$
- The marginal is distributed GAUSSian and is given by:

$$p(\boldsymbol{x}_a) = \int p(\boldsymbol{x}_a, \boldsymbol{x}_b)\, \mathrm{d}\boldsymbol{x}_b = \mathcal{N}\left(\boldsymbol{x}_a | \boldsymbol{\mu}_a, \boldsymbol{\Sigma}_{aa}\right) \tag{7}$$

- Intuitively, we ignore everything we are not interested in, i. e. we integrate out $\boldsymbol{x}_b$ from the joint distribution $p(\boldsymbol{x}_a, \boldsymbol{x}_b)$ given by equation (3)
- The corresponding results hold for $p(\boldsymbol{x}_b)$ which is obtained by marginalization of $p(\boldsymbol{x}_a, \boldsymbol{x}_b)$ over $\boldsymbol{x}_a$

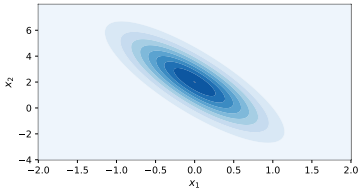
# Example: Marginal Gaussian Distribution

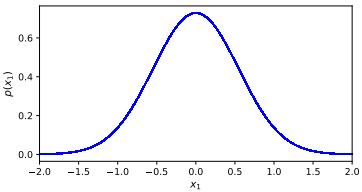
- Consider again

$$p(x_1, x_2) = \mathcal{N}\left(\begin{bmatrix}0\\2\end{bmatrix}, \begin{bmatrix}0.3 & -1\\-1 & 5\end{bmatrix}\right)$$

- According to equation (7), the marginal distribution $p(x_1)$ is given by:

$$p(x_1) = \mathcal{N}\big(0, 0.3\big)$$



# BAYES' Theorem for GAUSSian Variables

- Let a marginal GAUSSian distribution for $\boldsymbol{x}$ and a conditional GAUSSian distribution for $\boldsymbol{y}$ given $\boldsymbol{x}$ be given:

$$p(\boldsymbol{x}) := \mathcal{N}\left(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}\right) \qquad \text{and} \qquad p(\boldsymbol{y}|\boldsymbol{x}) := \mathcal{N}\left(\boldsymbol{y}; \boldsymbol{A}\boldsymbol{x}, \boldsymbol{\Lambda}\right) \tag{8}$$

- Then we obtain for the GAUSSian distributions $p(\boldsymbol{y})$ and $p(\boldsymbol{x}|\boldsymbol{y})$:

$$p(\boldsymbol{y}) = \mathcal{N}\left(\boldsymbol{y}; \boldsymbol{A}\boldsymbol{\mu}, \boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^\top + \boldsymbol{\Lambda}\right) \tag{9}$$

$$p(\boldsymbol{x}|\boldsymbol{y}) = \mathcal{N}\left(\boldsymbol{x}; \boldsymbol{\Gamma}\left[\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} + \boldsymbol{A}^\top \boldsymbol{\Lambda}^{-1}\boldsymbol{y}\right], \boldsymbol{\Gamma}\right) \tag{10}$$

$$\boldsymbol{\Gamma} := \left(\boldsymbol{\Sigma}^{-1} + \boldsymbol{A}^\top \boldsymbol{\Lambda}^{-1}\boldsymbol{A}\right)^{-1}$$

# Summary: Gaussian Distributions

- Gaussian distributions play a major role in machine learning
- Complex operations have **analytical solutions**
- Important operations on Gaussians (i. e. **conditioning**, **marginalization**, **convolution**) **result again in Gaussian distributions**
- Remember:

**Once a Gaussian, always a Gaussian!**

Section:

# BAYESian Linear Regression

Maximum Likelihood Estimation (MLE)
Maximum A Posteriori (MAP) Estimation
Full BAYESian Linear Regression

# Remember: A probabilistic View on Regression

- **Assumption 1:** The target values $y$ and the inputs $\boldsymbol{x}$ are related via the equation

$$y = h_{\boldsymbol{\theta}}(\boldsymbol{x}) + \varepsilon = \boldsymbol{\theta}^\top \boldsymbol{x} + \varepsilon, \tag{11}$$

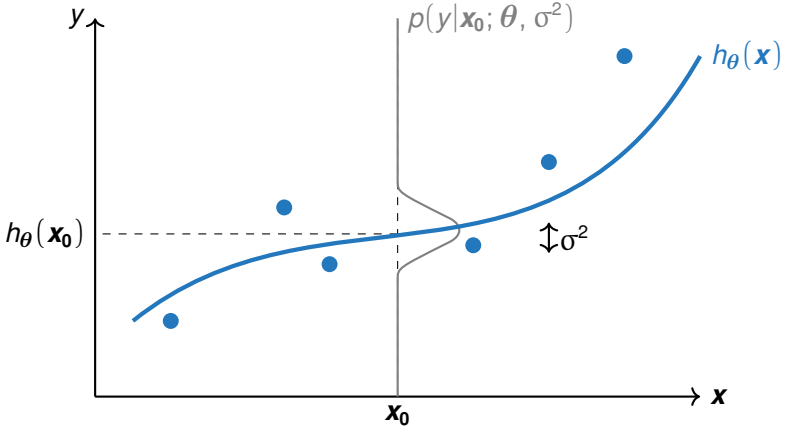
  where $\varepsilon$ is an error term which captures unmodeled effects or noise in the data
- **Assumption 2:** The noise $\varepsilon$ is a zero mean GAUSSian random variable

$$\varepsilon \sim \mathcal{N}(0, \sigma^2) \tag{12}$$

- We consider $y$ a random variable which is distributed according to

$$p(y|\boldsymbol{x}; \boldsymbol{\theta}, \sigma^2) = \mathcal{N}(y; h_{\boldsymbol{\theta}}(\boldsymbol{x}), \sigma^2) \tag{13}$$

# Remember: A probabilistic View on Regression (Ctd.)



# Remember: Likelihood Function for Regression

- We are given a training dataset $\mathcal{D} := \left\{ (\boldsymbol{x}^n, y_n) \right\}_{n=1}^N$
- The **(conditional) likelihood** of our regression model is given by:

$$p(\boldsymbol{y}|\boldsymbol{X}; \boldsymbol{\theta}, \sigma^2) = \prod_{n=1}^{N} \mathcal{N}\left(y_n; h_{\boldsymbol{\theta}}(\boldsymbol{x}^n), \sigma^2\right) = \prod_{n=1}^{N} \mathcal{N}\left(y_n; \boldsymbol{\theta}^\top \boldsymbol{\varphi}(\boldsymbol{x}^n), \sigma^2\right)$$

$$= \prod_{n=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} \cdot \exp\left( -\frac{1}{2\sigma^2} \left(y_n - \boldsymbol{\theta}^\top \boldsymbol{\varphi}(\boldsymbol{x}^n)\right)^2 \right) \tag{14}$$


# Remember: Likelihood Function for Regression (Ctd.)

- The **log-likelihood** is then given by *(we have computed this earlier already)*:

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}) &:= \log p(\boldsymbol{y}|\boldsymbol{X}; \boldsymbol{\theta}, \sigma^2) \\ &= -\frac{N}{2}\log(2\pi\sigma^2) - \frac{1}{\sigma^2} \cdot \underbrace{\frac{1}{2}\sum_{n=1}^{N}\left(y_n - \boldsymbol{\theta}^\top \boldsymbol{\varphi}(\boldsymbol{x}^n)\right)^2}_{\text{least squares error}} \end{aligned} \tag{15}$$

- We have to **minimize the least squares error** to **maximize the likelihood!**

**When minimizing the squared error we implicitly assume Gaussian noise!**


# Remember: The Maximum Likelihood Solution to Regression

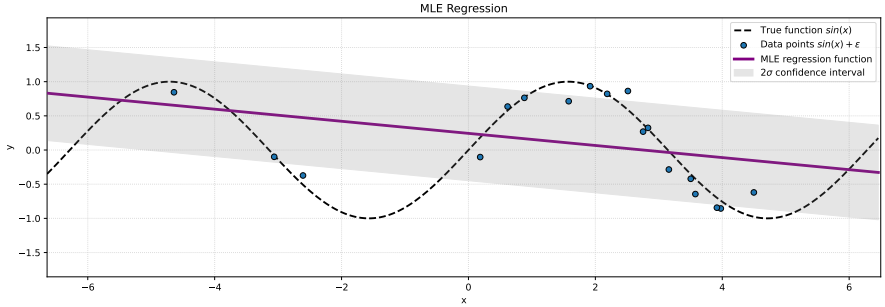
- Optimization of the log-likelihood function gives:

$$\boldsymbol{\theta}^{\mathrm{ML}} := (\boldsymbol{\Phi}^\top \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^\top \boldsymbol{y} \tag{16}$$

$$\left(\sigma^2\right)^{\mathrm{ML}} := \frac{1}{N} \sum_{n=1}^{N} \left(y_n - \left(\boldsymbol{\theta}^{\mathrm{ML}}\right)^\top \boldsymbol{\varphi}(\boldsymbol{x}^n)\right)^2 \tag{17}$$

- The probabilistic interpretation of linear regression allows us **to quantify the (global) uncertainty** of the model

# Example


MLE Regression
True function $sin(x)$
Data points $sin(x) + \epsilon$
MLE regression function
$2\sigma$ confidence interval
y
x
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−6
−4
−2
0
2
4
6


# Adding a Prior Distribution

- We have just seen that **maximum likelihood estimation is prone to overfitting**
- We now mitigate this effect by placing a **GAUSSian prior distribution**

$$p(\boldsymbol{\theta}; b^2) := \mathcal{N}(\mathbf{0}, b^2 \boldsymbol{I}) \tag{18}$$

on the parameters $\boldsymbol{\theta}$

- The prior distribution encodes what parameter values are plausible **(prior knowledge)** before having seen any data
- Instead of maximizing the likelihood, we seek parameters $\boldsymbol{\theta}^{\text{MAP}}$ which **maximize the posterior distribution** $p(\boldsymbol{\theta}|\mathcal{D})$ given a dataset $\mathcal{D}$

# Adding a Prior Distribution (Ctd.)

**Question:** Why do we use GAUSSians to model the prior?

- We set $p(\boldsymbol{\theta}; b^2) := \mathcal{N}(\mathbf{0}, b^2\boldsymbol{I})$
- The most important argument is **conjugacy**: Both, the prior and the likelihood are modelled by GAUSSian distributions. Therefore, the product of both – the posterior distribution – will also be GAUSSian *(only few distributions have that property)*
- There are **analytical solutions to complex operations** on GAUSSians which again result in GAUSSian distributions *(see first section for details)*


# Recall BAYES' Theorem

In the following we shall often refer to **BAYES' theorem**:

**BAYES' Theorem:**

$$p(\boldsymbol{\theta}|\mathcal{D}) = \frac{\overbrace{p(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{\theta};\sigma^2)}^{\textit{likelihood}} \cdot \overbrace{p(\boldsymbol{\theta};b^2)}^{\textit{prior}}}{p(\boldsymbol{y}|\boldsymbol{X})} \propto p(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{\theta};\sigma^2) \cdot p(\boldsymbol{\theta};b^2) \tag{19}$$

**Remarks:**

- Until now we have maximized the likelihood without considering $p(\boldsymbol{\theta};b^2)$
- $\boldsymbol{\theta}$ is now a random variable!

# Log-Posterior Distribution

- We begin by computing the **log-posterior**:

$$\log p(\boldsymbol{\theta}|\mathcal{D}) = \underbrace{\log p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta}; \sigma^2)}_{\text{log-likelihood}} + \underbrace{\log p(\boldsymbol{\theta}; b^2)}_{\text{log-prior}} + \text{const} \tag{20}$$

- The constant 'const' contains all terms independent of $\boldsymbol{\theta}$
- The MAP (*maximum aposteriori*) estimate will be a **compromise** between the parameter prior and the data-dependent likelihood
- As usual, we minimize the **negative log-posterior**

$$\boldsymbol{\theta}^{\text{MAP}} := \arg\min_{\boldsymbol{\theta}} \left\{ -\log p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta}; \sigma^2) - \log p(\boldsymbol{\theta}; b^2) \right\} \tag{21}$$

# Rewriting the Log-Likelihood Function for Regression

**The log-likelihood in equation** (15) **can be rewritten in matrix-vector notation:**

- The sum in the second term can be rewritten as:

$$\sum_{n=1}^{N} \left(y_n - \boldsymbol{\theta}^\top \boldsymbol{\varphi}(\boldsymbol{x}^n)\right)^2 \quad \Longleftrightarrow \quad (\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\theta})^\top (\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\theta}) \tag{22}$$

- Also, we notice that the first term $-\frac{N}{2}\log(2\pi\sigma^2)$ is constant with respect to $\boldsymbol{\theta}$
- Thus,

$$\log p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta}; \sigma^2) = -\frac{1}{2\sigma^2}(\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\theta})^\top (\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\theta}) + \text{ const} \tag{23}$$

# Rewriting the negative Log-Posterior

**Negative log-posterior:**

$$
\begin{aligned}
-\log p(\boldsymbol{\theta}|\mathcal{D}) &= -\log p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta}; \sigma^2) - \log p(\boldsymbol{\theta}; b^2) + \text{ const} \\
&= \frac{1}{2\sigma^2}\big(\boldsymbol{y} - \boldsymbol{\Phi\theta}\big)^\top\big(\boldsymbol{y} - \boldsymbol{\Phi\theta}\big) + \frac{1}{2b^2}\boldsymbol{\theta}^\top\boldsymbol{\theta} + \text{ const} \\
&= \frac{1}{2\sigma^2}\big(\boldsymbol{y}^\top\boldsymbol{y} - \boldsymbol{y}^\top\boldsymbol{\Phi\theta} - (\boldsymbol{\Phi\theta})^\top\boldsymbol{y} + (\boldsymbol{\Phi\theta})^\top\boldsymbol{\Phi\theta}\big) + \frac{1}{2b^2}\boldsymbol{\theta}^\top\boldsymbol{\theta} + \text{ const} \\
&= \frac{1}{2\sigma^2}\big(\boldsymbol{\theta}^\top\boldsymbol{\Phi}^\top\boldsymbol{\Phi\theta} - 2\boldsymbol{\theta}^\top\boldsymbol{\Phi}^\top\boldsymbol{y}\big) + \frac{1}{2b^2}\boldsymbol{\theta}^\top\boldsymbol{\theta} + \text{ const} \qquad (24)
\end{aligned}
$$

# Remember: Transposition Rules and Vector Derivatives

Remember the following rules:

**Matrix transposition rules:**

$$(\boldsymbol{A}^\top)^\top = \boldsymbol{A} \tag{25}$$

$$(\boldsymbol{A} + \boldsymbol{B})^\top = \boldsymbol{A}^\top + \boldsymbol{B}^\top \tag{26}$$

$$(\boldsymbol{A}\boldsymbol{B})^\top = \boldsymbol{B}^\top \boldsymbol{A}^\top \tag{27}$$

**Vector derivatives:**

$$\frac{\partial}{\partial \boldsymbol{x}} \boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} = 2\boldsymbol{A}\boldsymbol{x} \tag{28}$$

**[Equation** (28) **only holds if *A* is symmetric]**

$$\frac{\partial}{\partial \boldsymbol{x}} \boldsymbol{a}^\top \boldsymbol{x} = \boldsymbol{a} \tag{29}$$

# Computation of the Gradient

**Gradient of the negative log-posterior:**

$$\begin{aligned}\frac{\partial}{\partial \boldsymbol{\theta}} - \log p(\boldsymbol{\theta}|\mathcal{D}) &= \frac{\partial}{\partial \boldsymbol{\theta}} \frac{1}{2\sigma^2}\left(\boldsymbol{\theta}^\top \boldsymbol{\Phi}^\top \boldsymbol{\Phi}\boldsymbol{\theta} - 2\boldsymbol{\theta}^\top \boldsymbol{\Phi}^\top \boldsymbol{y}\right) + \frac{\partial}{\partial \boldsymbol{\theta}} \frac{1}{2b^2}\boldsymbol{\theta}^\top \boldsymbol{\theta} \\ &= \frac{1}{\sigma^2}\left(\boldsymbol{\Phi}^\top \boldsymbol{\Phi}\boldsymbol{\theta} - \boldsymbol{\Phi}^\top \boldsymbol{y}\right) + \frac{1}{b^2}\boldsymbol{\theta} \end{aligned} \tag{30}$$

# Computation of the MAP Estimate

**Set gradient to zero:**

$$\frac{1}{\sigma^2}\left(\boldsymbol{\Phi}^\top\boldsymbol{\Phi}\boldsymbol{\theta} - \boldsymbol{\Phi}^\top\boldsymbol{y}\right) + \frac{1}{b^2}\boldsymbol{\theta} \stackrel{!}{=} \mathbf{0}$$

$$\iff \left(\boldsymbol{\Phi}^\top\boldsymbol{\Phi} + \frac{\sigma^2}{b^2}\boldsymbol{I}\right)\boldsymbol{\theta} = \boldsymbol{\Phi}^\top\boldsymbol{y}$$

$$\iff \boldsymbol{\theta}^{\text{MAP}} = \left(\boldsymbol{\Phi}^\top\boldsymbol{\Phi} + \frac{\sigma^2}{b^2}\boldsymbol{I}\right)^{-1}\boldsymbol{\Phi}^\top\boldsymbol{y} \tag{31}$$

**Remark:** The matrix $\boldsymbol{\Phi}^\top\boldsymbol{\Phi} + \frac{\sigma^2}{b^2}\boldsymbol{I}$ is positive definite

# Matrix Invertibility

**Lemma:** The matrix $\boldsymbol{\Phi}^\top \boldsymbol{\Phi} + \frac{\sigma^2}{b^2} \boldsymbol{I}$ is positive definite (and therefore invertible).

**Proof:** Let $\boldsymbol{z} \in \mathbb{R}^M \backslash \{\boldsymbol{0}\}$ be a non-zero vector. We obtain

$$\boldsymbol{z}^\top \left( \boldsymbol{\Phi}^\top \boldsymbol{\Phi} + \frac{\sigma^2}{b^2} \boldsymbol{I} \right) \boldsymbol{z} = \boldsymbol{z}^\top \boldsymbol{\Phi}^\top \boldsymbol{\Phi} \boldsymbol{z} + \frac{\sigma^2}{b^2} \boldsymbol{z}^\top \boldsymbol{I} \boldsymbol{z} > 0 \tag{32}$$

due to the positive semidefiniteness of $\boldsymbol{\Phi}^\top \boldsymbol{\Phi}$, the positive definiteness of $\boldsymbol{I}$, and the positivity of $\sigma^2$ and $b^2$. ■

# Comparison of MLE and MAP Estimates

**MLE Estimate:**

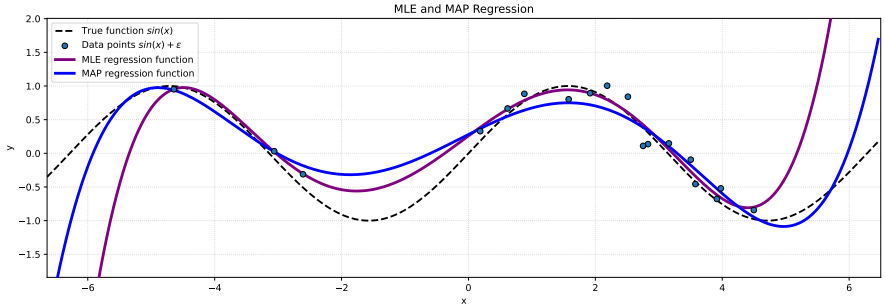
$$\boldsymbol{\theta}^{\mathrm{ML}} := \left(\boldsymbol{\Phi}^\top \boldsymbol{\Phi}\right)^{-1} \boldsymbol{\Phi}^\top \boldsymbol{y} \tag{33}$$

**MAP Estimate:**

$$\boldsymbol{\theta}^{\mathrm{MAP}} := \left(\boldsymbol{\Phi}^\top \boldsymbol{\Phi} + \frac{\sigma^2}{b^2} \boldsymbol{I}\right)^{-1} \boldsymbol{\Phi}^\top \boldsymbol{y} \tag{34}$$

For $\lambda := \frac{\sigma^2}{b^2}$ the MAP estimate is identical to ridge regression!

# Comparison of MLE and MAP Estimates (Ctd.)

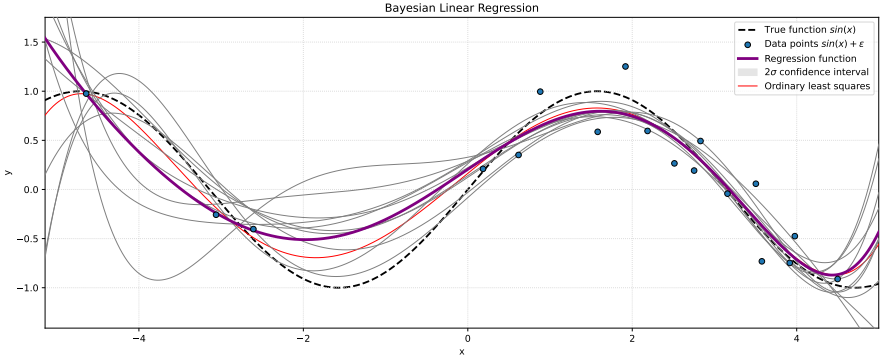



$\sigma^2 := 0.2$ and $b^2 := 0.1$


# Full BAYESian Regression

- So far we have seen that MLE estimates tend to overfit the training data
- In MAP estimation we place a prior on the parameters $\boldsymbol{\theta}$ which plays the role of a **regularizer**
- **BAYESian linear regression** pushes this idea to the next level:
  - Again we consider a prior distribution $p(\boldsymbol{\theta})$ of the parameters
  - But: We **do not** compute a point estimate of the parameters
  - Instead, we take the **full posterior distribution** into account when making predictions
  - **This means we do not fit any parameters, but we compute a mean over all plausible parameter settings** *(according to the posterior)*

# What we want to achieve:



# BAYESian Regression Model

In BAYESian linear regression we consider the following model:

$$\textbf{Prior} \qquad p(\boldsymbol{\theta}) := \mathcal{N}\big(\boldsymbol{\theta}; \boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0\big) \tag{35}$$

$$\textbf{Likelihood} \qquad p\big(y|\boldsymbol{x}', \boldsymbol{\theta}\big) := \mathcal{N}\big(y; \boldsymbol{\theta}^\top \boldsymbol{\varphi}(\boldsymbol{x}'), \sigma^2\big) \tag{36}$$

$\boldsymbol{x}'$ is a test data point, $\boldsymbol{\mu}_0$ is the mean of the prior distribution (usually we set $\boldsymbol{\mu}_0 := \mathbf{0}$), and $\boldsymbol{\Sigma}_0$ its covariance matrix


# Prior Predictions

- In practice, we are (usually) not interested in the parameter values themselves
- Instead, we want to **predict labels of unseen instances**
- Let $\boldsymbol{x}'$ be a test data point whose label $y'$ we wanto to predict
- The **prior predictive distribution** is given by

$$p(y|\boldsymbol{x}') = \int_{\boldsymbol{\theta}} p(y|\boldsymbol{x}', \boldsymbol{\theta}) p(\boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{\theta} = \mathbb{E}_{\boldsymbol{\theta}}\big\{p(y|\boldsymbol{x}', \boldsymbol{\theta})\big\} \tag{37}$$

- We can interpret equation (37) as the **average prediction** for all plausible parameter values according to the prior distribution


# Prior Predictions (Ctd.)

**Parameters of the (prior) predictive distribution:**

Due to the **conjugate GAUSSian prior** $p(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$, we can compute the parameters of the predictive distribution in **closed form**:

$$p(y|\boldsymbol{x}') = \mathcal{N}\big(y; \boldsymbol{\mu}, \boldsymbol{\Sigma}\big) \tag{38}$$

with

$$\boldsymbol{\mu} := \boldsymbol{\mu}_0^\top \boldsymbol{\varphi}(\boldsymbol{x}') \tag{39}$$

$$\boldsymbol{\Sigma} := \boldsymbol{\varphi}(\boldsymbol{x}')^\top \boldsymbol{\Sigma}_0 \boldsymbol{\varphi}(\boldsymbol{x}') + \sigma^2 \tag{40}$$

**Remark:** We have used equation (9) here

# Prior Predictions (Ctd.)

- In equation (40), the term $\boldsymbol{\varphi}(\boldsymbol{x}')^\top \boldsymbol{\Sigma}_0 \boldsymbol{\varphi}(\boldsymbol{x}')$ accounts for the **uncertainty** associated with the parameters $\boldsymbol{\theta}$
- $\sigma^2$ is the uncertainty contribution due to measurement noise
- Every $\boldsymbol{\theta}^i$ we sample from the prior distribution gives rise to a function

$$f_i(\cdot) := \left(\boldsymbol{\theta}^i\right)^\top \boldsymbol{\varphi}(\cdot) \tag{41}$$

- The parameter distribution $p(\boldsymbol{\theta})$ induces a distribution $p\big(f(\cdot)\big)$ over functions
- The prior predictions are made **without taking the training data into account**, therefore these functions do not fit the data well

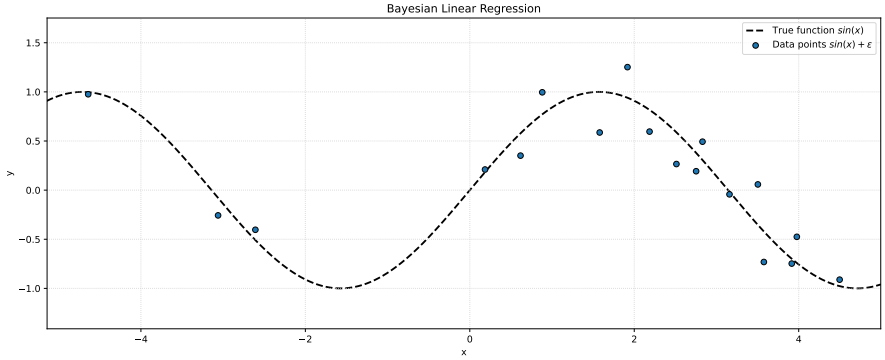
# Example: Setup

- Let us have a look at an example
- Suppose the labels are generated by the **true function** $f(x) = \sin(x)$ plus some additive noise $\varepsilon \sim \mathcal{N}(0, \sigma^2)$
- A training data point $(x_n, y_n)$ then takes the form

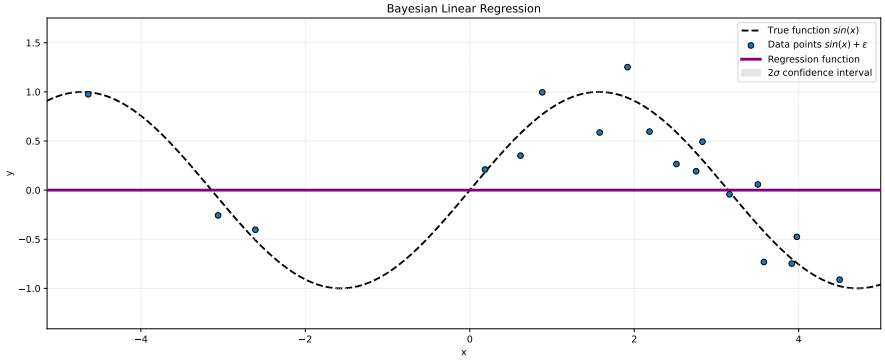
$$(x_n, y_n) = \big(x_n, \sin(x_n) + \varepsilon_n\big) \qquad n = 1, 2, \ldots, N \tag{42}$$

- We initialize the **parameter prior distribution** to be $p(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}|\mathbf{0}, \boldsymbol{I})$, i. e. zero mean and **isotropic** *(i. e. rotation invariant)* covariance

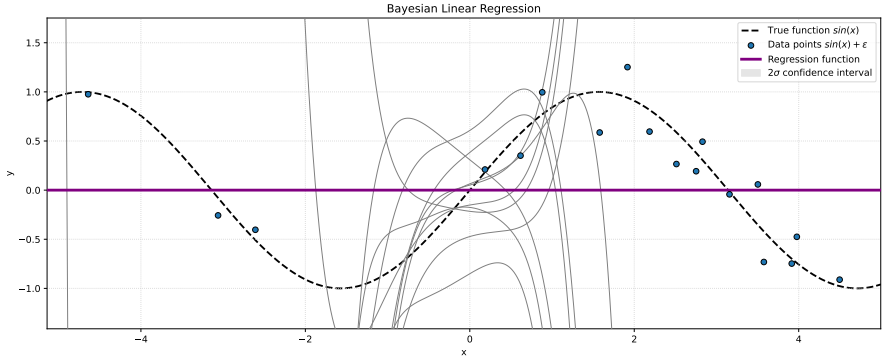
# Example: Dataset



# Example: Prior Predictions



# Example: Sampled Functions



# Incorporation of the Training Data

- So far we have used the **prior distribution** $p(\boldsymbol{\theta})$ when computing predictions
- In the following we will replace $p(\boldsymbol{\theta})$ with the **posterior distribution** $p(\boldsymbol{\theta}|\mathcal{D})$ of $\boldsymbol{\theta}$ given the training data $\mathcal{D}$
- We compute the posterior distribution according to BAYES' theorem analogously to equation (19):

$$p(\boldsymbol{\theta}|\mathcal{D}) = \frac{\overbrace{p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta})}^{\textit{likelihood}} \cdot \overbrace{p(\boldsymbol{\theta})}^{\textit{prior}}}{p(\boldsymbol{y}|\boldsymbol{X})} \tag{43}$$

- The parameter posterior is **again GAUSSian**


# Posterior Parameter Distribution

**Parameter posterior distribution:**

The parameter posterior distribution is given by

$$p(\boldsymbol{\theta}|\mathcal{D}) = \mathcal{N}\big(\boldsymbol{\theta}; \boldsymbol{\mu}_N, \boldsymbol{\Sigma}_N\big) \tag{44}$$

with

$$\boldsymbol{\Sigma}_N := \big(\boldsymbol{\Sigma}_0^{-1} + \sigma^{-2}\boldsymbol{\Phi}^\top\boldsymbol{\Phi}\big)^{-1} \tag{45}$$

$$\boldsymbol{\mu}_N := \boldsymbol{\Sigma}_N\big(\boldsymbol{\Sigma}_0^{-1}\boldsymbol{\mu}_0 + \sigma^{-2}\boldsymbol{\Phi}^\top\boldsymbol{y}\big) \tag{46}$$

**Remark:** We have used equation (10) here

# Posterior Predictions (Ctd.)

## Parameters of the (posterior) predictive distribution:

The parameters of the posterior predictive distribution are computed analogously to those of the prior predictive distribution. We replace $\boldsymbol{\mu}_0$ and $\boldsymbol{\Sigma}_0$ with $\boldsymbol{\mu}_N$ and $\boldsymbol{\Sigma}_N$, respectively:

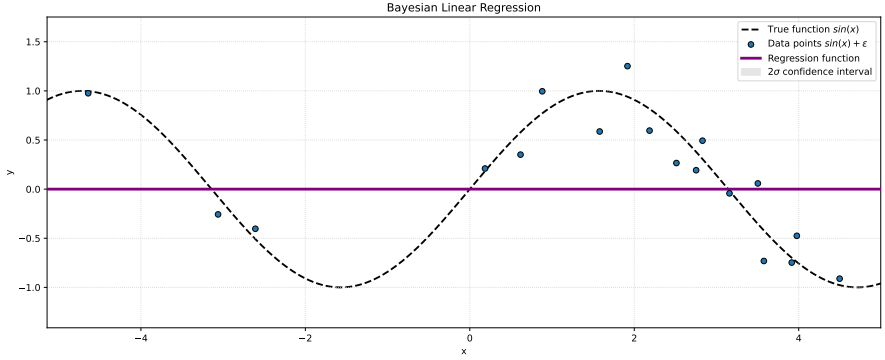
$$p(y|\boldsymbol{x}') = \mathcal{N}\left(\boldsymbol{\mu}, \boldsymbol{\Sigma}\right) \tag{47}$$

with

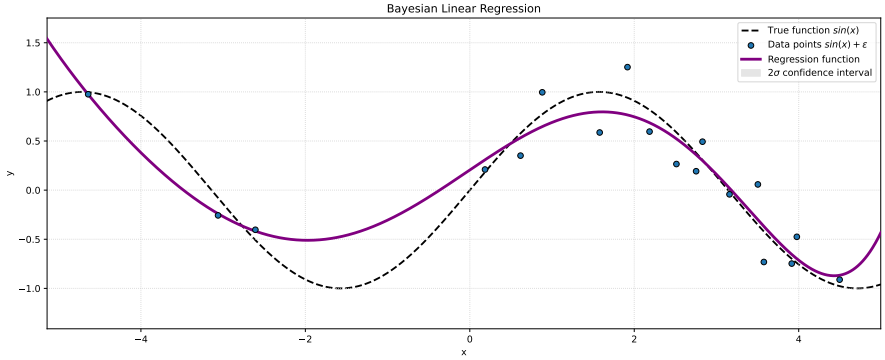
$$\boldsymbol{\mu} := \boldsymbol{\mu}_N^\top \boldsymbol{\varphi}(\boldsymbol{x}') \tag{48}$$

$$\boldsymbol{\Sigma} := \boldsymbol{\varphi}(\boldsymbol{x}')^\top \boldsymbol{\Sigma}_N \boldsymbol{\varphi}(\boldsymbol{x}') + \sigma^2 \tag{49}$$

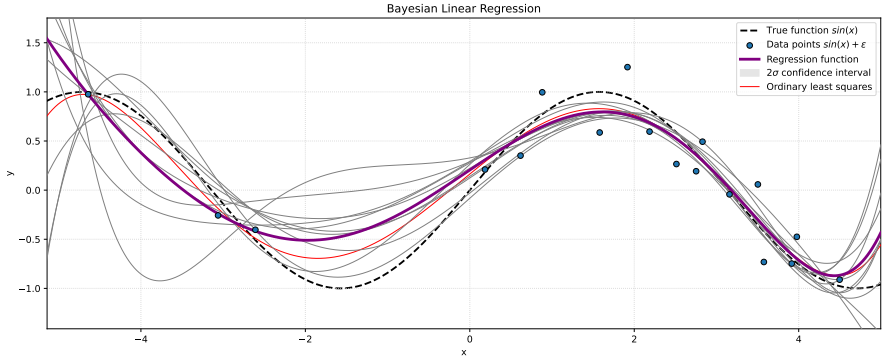
# Example: From Prior Predictions to Posterior Predictions



# Example: From Prior Predictions to Posterior Predictions (Ctd.)



# Example: From Prior Predictions to Posterior Predictions (Ctd.)

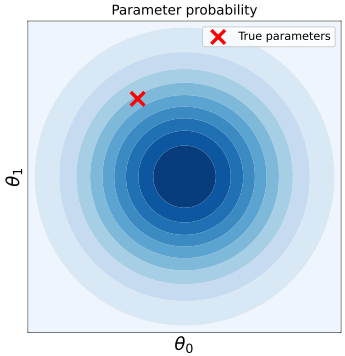


# Illustration of BAYESian Learning

- We sample data points from the function

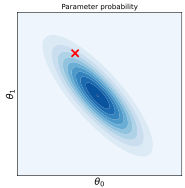
$$f(x) = 0.5x - 0.3 + \varepsilon$$

- $\varepsilon$ is additive noise with mean $\mu_\varepsilon = 0$ and variance $\sigma_\varepsilon^2 = 1$
- We place a **prior** on the parameters: $p(\boldsymbol{\theta}) := \mathcal{N}\left(\mathbf{0}, \boldsymbol{I}_2\right)$

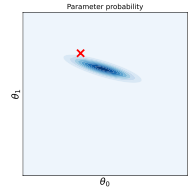


# Illustration of BAYESian Learning (Ctd.)

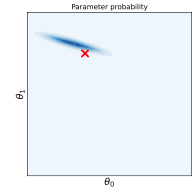
$N = 1$:




$N = 2$:

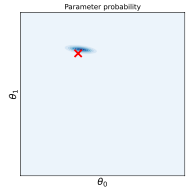


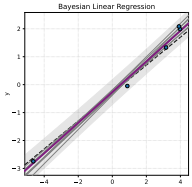

$N = 3$:



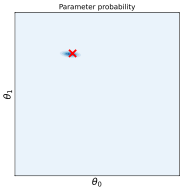
# Illustration of Bayesian Learning (Ctd.)

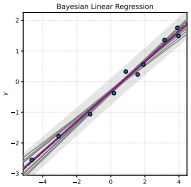
$N = 5$:



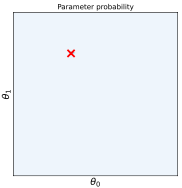



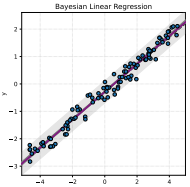

$N = 10$:




$N = 100$:



Section:

# Kernel Ridge Regression

Introduction
Woodbury Matrix Identity
Derivation of the Algorithm
Example

# Introduction of Kernels for Regression

- We now turn to a **kernel approach** to regression.
- Remember the **ridge regression** formula:

$$\boldsymbol{\theta} = \left(\boldsymbol{\Phi}^\top \boldsymbol{\Phi} + \lambda \boldsymbol{I}\right)^{-1} \boldsymbol{\Phi}^\top \boldsymbol{y}, \tag{50}$$

  where $\lambda \geqslant 0$ is the regularization parameter
- Recall: In order to apply kernels, we have to rephrase this equation in terms of **dot products of the input features**

# A useful Matrix Identity

**Woodbury matrix identity / 'push-through' identity:**

The following identity holds for square matrices $\boldsymbol{P}$ and $\boldsymbol{Q}$:

$$(\boldsymbol{PQ} + \boldsymbol{I})^{-1}\boldsymbol{P} = \boldsymbol{P}(\boldsymbol{QP} + \boldsymbol{I})^{-1} \tag{51}$$

**Proof:** Multiply the the equation by $(\boldsymbol{PQ} + \boldsymbol{I})$ from the left, and by $(\boldsymbol{QP} + \boldsymbol{I})$ from the right. The resulting equation is $\boldsymbol{P}(\boldsymbol{QP} + \boldsymbol{I}) = (\boldsymbol{PQ} + \boldsymbol{I})\boldsymbol{P}$. This equation is straightforward to verify by performing the multiplications according to the distributive law. ∎

# Derivation of Kernel Ridge Regression

- We apply equation (51) to the matrix $\left(\boldsymbol{\Phi}^\top \boldsymbol{\Phi} + \lambda \boldsymbol{I}\right)^{-1} \boldsymbol{\Phi}^\top$
- For this we set:

$$\boldsymbol{P} := \boldsymbol{\Phi}^\top \qquad \boldsymbol{Q} := \boldsymbol{\Phi}$$

- We receive:

$$\left(\boldsymbol{\Phi}^\top \boldsymbol{\Phi} + \lambda \boldsymbol{I}\right)^{-1} \boldsymbol{\Phi}^\top = \boldsymbol{\Phi}^\top \left(\boldsymbol{\Phi} \boldsymbol{\Phi}^\top + \lambda \boldsymbol{I}\right)^{-1} \tag{52}$$

- We plug this result into equation (50) to obtain

$$\boldsymbol{\theta} = \left(\boldsymbol{\Phi}^\top \boldsymbol{\Phi} + \lambda \boldsymbol{I}\right)^{-1} \boldsymbol{\Phi}^\top \boldsymbol{y} = \boldsymbol{\Phi}^\top \left(\boldsymbol{\Phi} \boldsymbol{\Phi}^\top + \lambda \boldsymbol{I}\right)^{-1} \boldsymbol{y} \tag{53}$$


# Derivation of Kernel Ridge Regression (Ctd.)

To predict the label $y'$ of an unseen instance $\boldsymbol{x}'$ we compute:

$$\begin{aligned} y' &= \boldsymbol{\theta}^\top \boldsymbol{\varphi}(\boldsymbol{x}') \\ &= \left(\boldsymbol{\Phi}^\top \left(\boldsymbol{\Phi}\boldsymbol{\Phi}^\top + \lambda \boldsymbol{I}\right)^{-1} \boldsymbol{y}\right)^\top \boldsymbol{\varphi}(\boldsymbol{x}') \\ &= \boldsymbol{y}^\top \left(\boldsymbol{\Phi}\boldsymbol{\Phi}^\top + \lambda \boldsymbol{I}\right)^{-1} \boldsymbol{\Phi}\boldsymbol{\varphi}(\boldsymbol{x}') \end{aligned}$$

[The features appear exclusively in terms of inner products – **we can apply kernels!**]

$$= \boxed{\boldsymbol{y}^\top \left(\boldsymbol{K} + \lambda \boldsymbol{I}\right)^{-1} \boldsymbol{\kappa}(\boldsymbol{x}')} \tag{54}$$

# Derivation of Kernel Ridge Regression (Ctd.)

In equation (54) we have set:

$$\boldsymbol{\Phi}\boldsymbol{\varphi}(\boldsymbol{x}') = \begin{pmatrix} \boldsymbol{\varphi}(\boldsymbol{x}^1)^\top \\ \boldsymbol{\varphi}(\boldsymbol{x}^2)^\top \\ \vdots \\ \boldsymbol{\varphi}(\boldsymbol{x}^N)^\top \end{pmatrix} \boldsymbol{\varphi}(\boldsymbol{x}') = \begin{pmatrix} \boldsymbol{\varphi}(\boldsymbol{x}^1)^\top \boldsymbol{\varphi}(\boldsymbol{x}') \\ \boldsymbol{\varphi}(\boldsymbol{x}^2)^\top \boldsymbol{\varphi}(\boldsymbol{x}') \\ \vdots \\ \boldsymbol{\varphi}(\boldsymbol{x}^N)^\top \boldsymbol{\varphi}(\boldsymbol{x}') \end{pmatrix} = \begin{pmatrix} k(\boldsymbol{x}^1, \boldsymbol{x}') \\ k(\boldsymbol{x}^2, \boldsymbol{x}') \\ \vdots \\ k(\boldsymbol{x}^N, \boldsymbol{x}') \end{pmatrix} =: \boldsymbol{\kappa}(\boldsymbol{x}')$$

The vector $\boldsymbol{\kappa}(\boldsymbol{x}')$ contains the pairwise similarities of $\boldsymbol{x}'$ and all training data points, i. e. **it must be computed from scratch each time we want to do inference!**

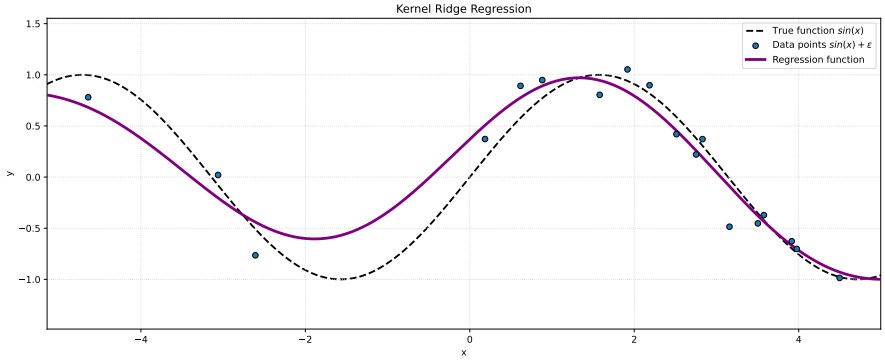
# Derivation of Kernel Ridge Regression (Ctd.)

$$\boldsymbol{\Phi}\boldsymbol{\Phi}^\top = \begin{pmatrix} \varphi(\boldsymbol{x}^1)^\top \\ \vdots \\ \varphi(\boldsymbol{x}^N)^\top \end{pmatrix} \begin{pmatrix} \varphi(\boldsymbol{x}^1)^\top \\ \vdots \\ \varphi(\boldsymbol{x}^N)^\top \end{pmatrix}^\top = \begin{pmatrix} \varphi(\boldsymbol{x}^1)^\top\varphi(\boldsymbol{x}^1) & \dots & \varphi(\boldsymbol{x}^1)^\top\varphi(\boldsymbol{x}^N) \\ \vdots & \ddots & \vdots \\ \varphi(\boldsymbol{x}^N)^\top\varphi(\boldsymbol{x}^1) & \dots & \varphi(\boldsymbol{x}^N)^\top\varphi(\boldsymbol{x}^N) \end{pmatrix}$$

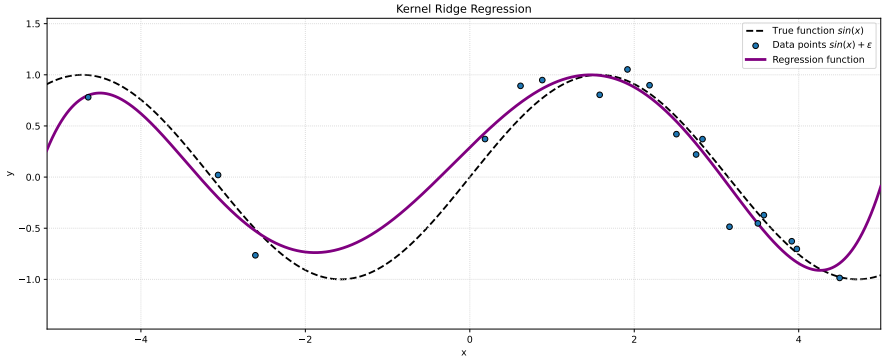
$$= \begin{pmatrix} k(\boldsymbol{x}^1, \boldsymbol{x}^1) & \dots & k(\boldsymbol{x}^1, \boldsymbol{x}^N) \\ \vdots & \ddots & \vdots \\ k(\boldsymbol{x}^N, \boldsymbol{x}^1) & \dots & k(\boldsymbol{x}^N, \boldsymbol{x}^N) \end{pmatrix} =: \boldsymbol{K} \tag{55}$$

Only the training data is needed to compute $\boldsymbol{K}$. **We can do it once and reuse it!**

# Example: Kernel Ridge Regression – Gaussian Kernel, s := 2.0



# Example: Kernel Ridge Regression – Polynomial Kernel, p := 5



Section:

# Gaussian Process Regression


Introduction
Predictions using noise-free Observations
Predictions using noisy Observations
Learning the Hyperparameters

# So far...

**So far:** In BAYESian linear regression we focused on **parametric representations** of the regression function $h$, i. e. we introduced a set of adjustable parameters $\boldsymbol{\theta}$ and assumed $h(\boldsymbol{x}) = \boldsymbol{\theta}^\top \boldsymbol{x}$ to be of a fixed parametric form.

We then inferred a posterior distribution $p(\boldsymbol{\theta}|\mathcal{D})$ over the parameters given the training data. We could use the posterior to predict the label of unseen instances.

# What is a GAUSSian Process?

**Now:** A **GAUSSian process (GP)** model extends on the **non-parametric** kernel-approach used in kernel ridge regression and performs **BAYESian inference over functions themselves**. This means that there are no parameters $\boldsymbol{\theta}$ involved that need to be learned from training data. Similarly to BAYESian regression, GPs produce a probabilistic output.

Formally, a GAUSSian process is a collection of random variables, any finite number of which has a **joint GAUSSian distribution**.

# GAUSSian Processes for Regression

- In this section we introduce GPs for regression
- Let the **prior** on the regression function $h$ be a GP, denoted by

$$y = h(\boldsymbol{x}) \sim \mathrm{GP}\big(m(\boldsymbol{x}), k(\boldsymbol{x}, \widetilde{\boldsymbol{x}})\big), \tag{56}$$

where $m(\boldsymbol{x})$ is the **mean function** and $k(\boldsymbol{x}, \widetilde{\boldsymbol{x}})$ the positive semi-definite kernel function **(covariance function)**, i. e.

$$m(\boldsymbol{x}) = \mathbb{E}\big\{h(\boldsymbol{x})\big\} \tag{57}$$

$$k(\boldsymbol{x}, \widetilde{\boldsymbol{x}}) = \mathbb{E}\big\{\big(h(\boldsymbol{x}) - m(\boldsymbol{x})\big)\big(h(\widetilde{\boldsymbol{x}}) - m(\widetilde{\boldsymbol{x}})\big)^\top\big\} \tag{58}$$

# GAUSSian Processes for Regression (Ctd.)

For any **finite set** of points $\boldsymbol{x}^1, \ldots, \boldsymbol{x}^N$, this GAUSSian process defines a **joint GAUSSian** distribution:

$$p(\boldsymbol{y}|\boldsymbol{X}) = \mathcal{N}\big(\boldsymbol{y}; \boldsymbol{\mu}, \boldsymbol{K}\big) \tag{59}$$

where

$$\boldsymbol{\mu} := \big(m(\boldsymbol{x}^1), \ldots, m(\boldsymbol{x}^N)\big)^\top \tag{60}$$

$$\boldsymbol{K} := \big[k_{ij}\big]_{i,j=1,\ldots,N} \qquad \text{with} \qquad k_{ij} := k(\boldsymbol{x}^i, \boldsymbol{x}^j) \tag{61}$$

# Noise-free Observations: Training Setup

- Suppose we observe a training set

$$\mathcal{D} := \left\{ (\boldsymbol{x}^n, y_n) \right\}_{n=1}^{N}$$

  where we assume $y_n$ to be a **noise-free** observation
- Given a set of $N_*$ test features $\boldsymbol{X}_*$ we want to predict the respective labels $\boldsymbol{y}_*$
- When assuming noise-free observations, a GP will act as an **interpolator** of the training data, i. e. it predicts the training data with **no uncertainty**

# Joint Distribution (noise-free)

By definition the **joint distribution** of training and test data takes the form of a **partitioned Gaussian distribution** – compare with equation (3):

$$\begin{pmatrix} \boldsymbol{y} \\ \boldsymbol{y}_* \end{pmatrix} \sim \mathcal{N}\left( \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix}, \begin{bmatrix} \boldsymbol{K} & \boldsymbol{K}_* \\ \boldsymbol{K}_*^\top & \boldsymbol{K}_{**} \end{bmatrix} \right), \tag{62}$$

where:

$$\boldsymbol{K} := k(\boldsymbol{X}, \boldsymbol{X}) \in \mathbb{R}^{N \times N}$$
$$\boldsymbol{K}_* := k(\boldsymbol{X}, \boldsymbol{X}_*) \in \mathbb{R}^{N \times N_*}$$
$$\boldsymbol{K}_{**} := k(\boldsymbol{X}_*, \boldsymbol{X}_*) \in \mathbb{R}^{N_* \times N_*}$$


# GP Posterior Distribution (noise-free)

## GP posterior distribution (noise-free observations)

Using rules (5) and (6) for conditioning Gaussians, we get

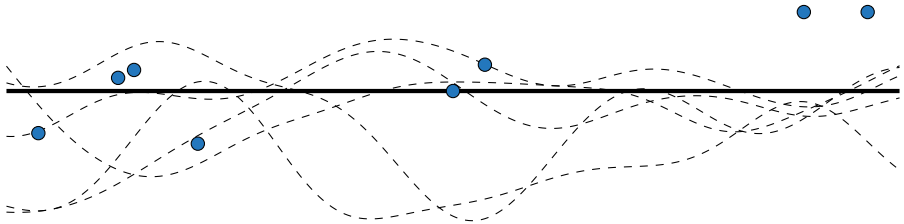
$$p(\boldsymbol{y}_*|\boldsymbol{X}_*, \boldsymbol{X}, \boldsymbol{y}) = \mathcal{N}\left(\boldsymbol{y}_*; \boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*\right) \tag{63}$$

$$\boldsymbol{\mu}_* := \boldsymbol{y}^\top \boldsymbol{K}^{-1} \boldsymbol{K}_* \tag{64}$$

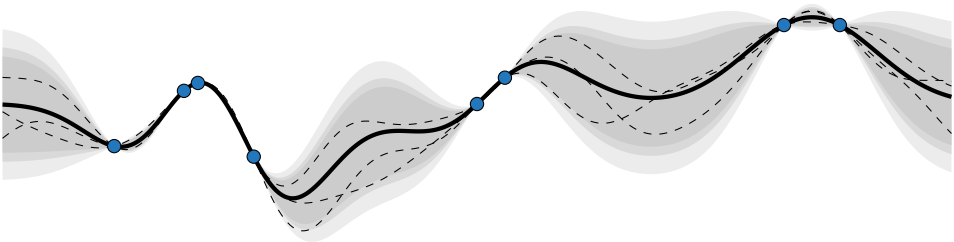
$$\boldsymbol{\Sigma}_* := \boldsymbol{K}_{**} - \boldsymbol{K}_*^\top \boldsymbol{K}^{-1} \boldsymbol{K}_* \tag{65}$$

**Remark:** $\boldsymbol{\Sigma}_*$ is called the **SCHUR complement**

# Example: GP Prior Distribution



# Example: GP Posterior Distribution (noise-free)



# Joint Distribution (noisy)

- Now let us consider the case where what we observe is a **noisy version** of the underlying function: $y = f(\boldsymbol{x}) + \varepsilon$, where $\varepsilon \sim \mathcal{N}(0, \sigma^2)$
- The covariance of the observed noisy data is given by:

$$\widetilde{\boldsymbol{K}} := \boldsymbol{K} + \sigma^2 \boldsymbol{I}_N \tag{66}$$

- Thus, the joint GAUSSian density becomes:

$$\begin{pmatrix} \boldsymbol{y} \\ \boldsymbol{y}_* \end{pmatrix} \sim \mathcal{N}\left( \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix}, \begin{bmatrix} \widetilde{\boldsymbol{K}} & \boldsymbol{K}_* \\ \boldsymbol{K}_*^\top & \boldsymbol{K}_{**} \end{bmatrix} \right) \tag{67}$$

# GP Posterior Distribution (noisy)

## GP posterior distribution (noisy observations)

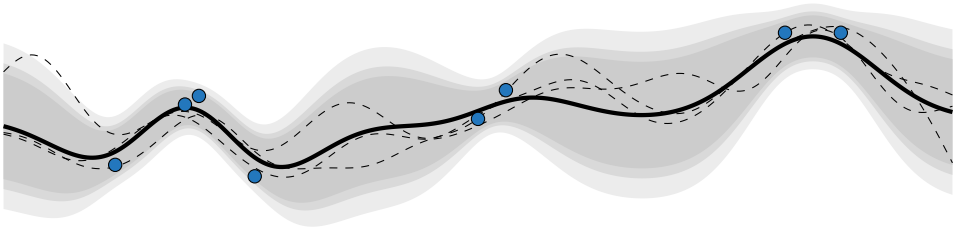
Using rules (5) and (6) for conditioning GAUSSians, we get

$$p(\boldsymbol{y}_*|\boldsymbol{X}_*, \boldsymbol{X}, \boldsymbol{y}) = \mathcal{N}\big(\boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*\big) \tag{68}$$

$$\boldsymbol{\mu}_* := \boldsymbol{y}^\top \widetilde{\boldsymbol{K}}^{-1} \boldsymbol{K}_* \tag{69}$$

$$\boldsymbol{\Sigma}_* := \boldsymbol{K}_{**} - \boldsymbol{K}_*^\top \widetilde{\boldsymbol{K}}^{-1} \boldsymbol{K}_* \tag{70}$$

# Example: GP Posterior Distribution (noisy)



# Marginal Likelihood

- To estimate the kernel parameters $\boldsymbol{\theta}$, we could use grid search
- Here we consider an **empirical Bayes approach** – in particular we will maximize the **marginal likelihood**:

$$\log p(\boldsymbol{y}|\boldsymbol{X}) = \underbrace{-\frac{1}{2}\boldsymbol{y}^\top \widetilde{\boldsymbol{K}}^{-1}\boldsymbol{y}}_{\text{❶}} \underbrace{-\frac{1}{2}\log\det \widetilde{\boldsymbol{K}}}_{\text{❷}} \underbrace{-\frac{N}{2}\log 2\pi}_{\text{❸}} \tag{71}$$

- Term ❶ is a **data fit term**, term ❷ is a **model complexity term**, and the last term ❸ is just a constant

# Gradient of the Marginal Likelihood

- To maximize expression (71), we have to compute its gradient
- The partial derivative with respect to the $j$-th kernel parameter is given by:

$$\begin{aligned}\frac{\partial}{\partial \theta_j} \log p(\boldsymbol{y}|\boldsymbol{X}) &= \frac{1}{2}\boldsymbol{y}^\top \widetilde{\boldsymbol{K}}^{-1}\frac{\partial \widetilde{\boldsymbol{K}}}{\partial \theta_j}\widetilde{\boldsymbol{K}}^{-1}\boldsymbol{y} - \frac{1}{2}\mathrm{tr}\left(\widetilde{\boldsymbol{K}}^{-1}\frac{\partial \widetilde{\boldsymbol{K}}}{\partial \theta_j}\right) \\ &= \frac{1}{2}\mathrm{tr}\left(\left(\boldsymbol{\alpha}\boldsymbol{\alpha}^\top - \widetilde{\boldsymbol{K}}^{-1}\right)\frac{\partial \widetilde{\boldsymbol{K}}}{\partial \theta_j}\right) \qquad (72)\end{aligned}$$

- In equation (72) we have set $\boldsymbol{\alpha} := \widetilde{\boldsymbol{K}}^{-1}\boldsymbol{y}$
- The expression $\frac{\partial \widetilde{\boldsymbol{K}}}{\partial \theta_j}$ depends on the kernel function used

# Some useful Rules

1. Trace of a matrix $\boldsymbol{A} \in \mathbb{R}^{D \times D}$ *(sum of the elements on the main diagonal)*:

$$\operatorname{tr}(\boldsymbol{A}) := \sum_{i=1}^{D} a_{ii} \tag{73}$$

2. Derivative of an inverse:

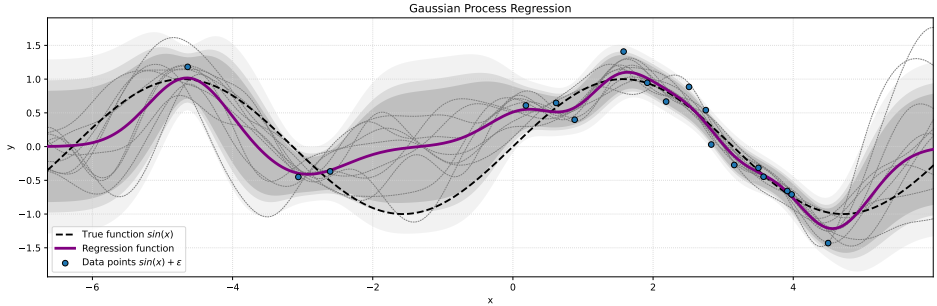
$$\frac{\partial \boldsymbol{A}^{-1}}{\partial x} = -\boldsymbol{A}^{-1} \frac{\partial \boldsymbol{A}}{\partial x} \boldsymbol{A}^{-1} \tag{74}$$

3. Derivative of a determinant:

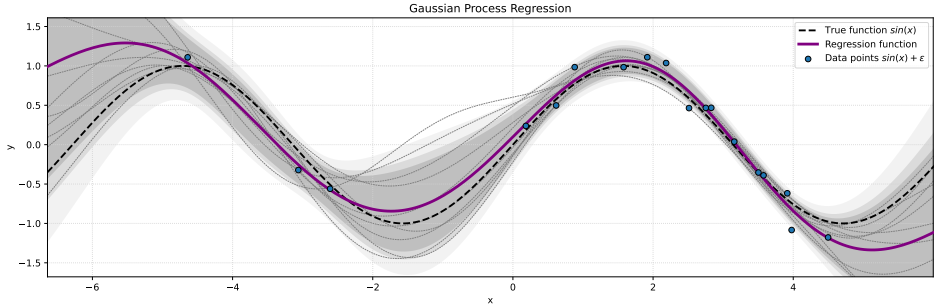
$$\frac{\partial \det \boldsymbol{A}}{\partial x} = \det(\boldsymbol{A}) \cdot \operatorname{tr}\left(\boldsymbol{A}^{-1} \frac{\partial \boldsymbol{A}}{\partial x}\right) \tag{75}$$

4. $\boldsymbol{a}^\top \boldsymbol{a} = \operatorname{tr}(\boldsymbol{a}\boldsymbol{a}^\top)$

# Example: Gaussian Process – not optimized



# Example: Gaussian Process – optimized



Section:

# Wrap-Up

Summary
Recommended Literature
Self-Test Questions

# Summary

- The maximum likelihood (MLE) estimate is **prone to overfitting**
- We can mitigate this issue by putting a **prior on the parameters** which leads to a maximum aposteriori (MAP) estimate
- In BAYESian regression we sum over all possible models and obtain a model average
- BAYESian regression also models the **local uncertainty**
- Kernel regression uses kernels to get rid of the parameters
- A GAUSSian process is a non-parametric version of BAYESian regression

# Recommended Literature

1. **MLE and MAP regression:**
   - [Deisenroth.2019], chapter 9
2. **Gaussian process regression:**
   - [Rasmussen/Williams.2006]
   - [Bishop.2006], chapter 6.4

(For free PDF versions, see list in GitHub readme!)


# Self-Test Questions

1. How is the least squares error related to the MLE estimate?
2. What is the difference between the MLE estimate and the MAP estimate?
3. Explain what BAYESian regression does? How does it work?
4. What is the prerequisite for using kernels in regression?
5. What are advantages / disadvantages of the kernel method?
6. What is a GAUSSian process?
7. How can we learn suitable hyperparameters?

# Thank you very much for the attention!

*** * * Artificial Intelligence and Machine Learning * * ***

**Topic:** Advanced Regression Techniques

**Term:** Summer term 2025

**Contact:**
Daniel Wehner, M.Sc.
SAP SE / DHBW Mannheim
daniel.wehner@sap.com

## Do you have any questions?