

*** Applied Machine Learning Fundamentals ***

Mathematical Foundations

Daniel Wehner, M.Sc.

SAP SE / DHBW Mannheim

Winter term 2023/2024



Find all slides on [GitHub](#) (DaWe1992/Applied_ML_Fundamentals)

Lecture Overview

- Unit I** Machine Learning Introduction
- Unit II** Mathematical Foundations
- Unit III** Bayesian Decision Theory
- Unit IV** Regression
- Unit V** Classification I
- Unit VI** Evaluation
- Unit VII** Classification II
- Unit VIII** Clustering
- Unit IX** Dimensionality Reduction

Agenda for this Unit

① Introduction

② Linear Algebra

Vectors

Matrices

Eigenvectors and Eigenvalues

Miscellaneous

③ Statistics

Random Variables and Common Distributions

Basic Rules of Probability

Expectation and Variance

Kullback-Leibler Divergence

④ Optimization

Introduction

Cost Functions and Convexity

Constrained Optimization and Lagrange

Multipliers

Numerical Optimization

⑤ Wrap-Up

Summary

Self-Test Questions

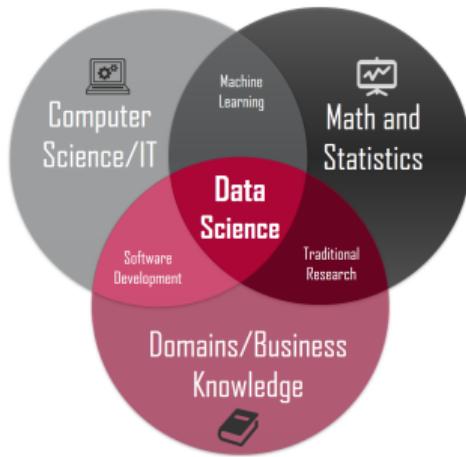
Lecture Outlook

Section: Introduction



Introduction

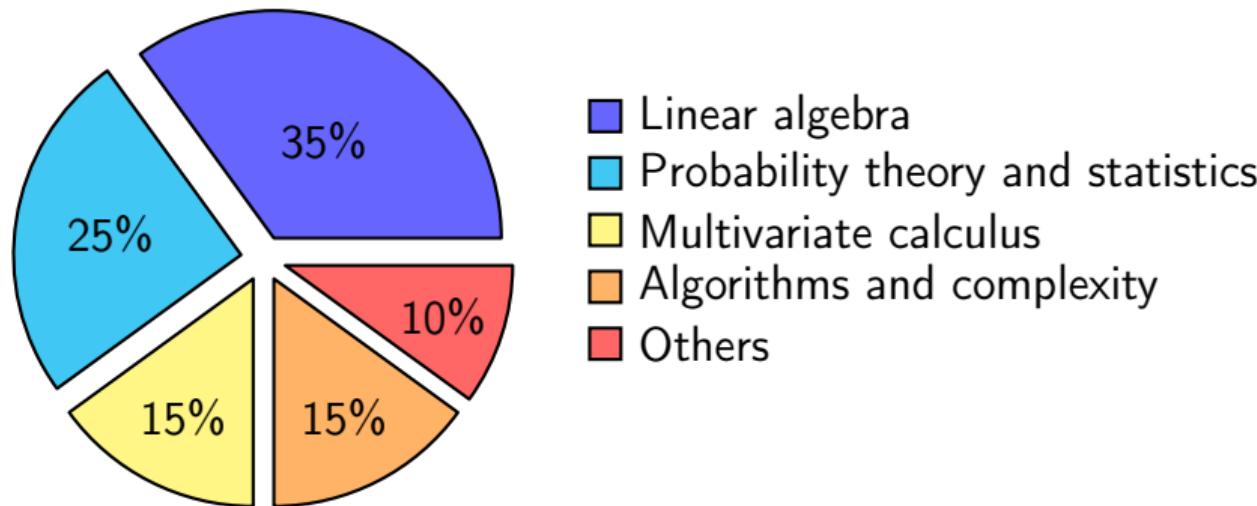
Math is a significant portion of data science / machine learning!



You will need it to understand:

- **Statistical** machine learning
- How optimization for learning / empirical risk minimization works
- How linear algebra, calculus and statistics are used to make learning and inference more efficient

Math is important!



Section:
Linear Algebra

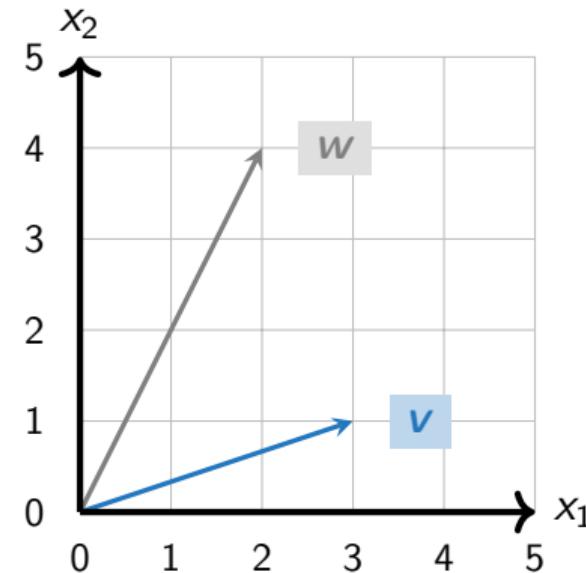


What is a Vector?

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$\mathbf{v} = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$$

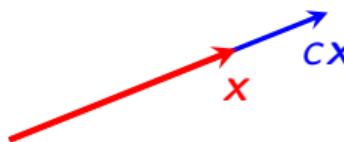
$$\mathbf{w} = \begin{bmatrix} 2 \\ 4 \end{bmatrix}$$



Multiplication by a Scalar

$$c\mathbf{x} = c \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} cx_1 \\ cx_2 \end{bmatrix}$$

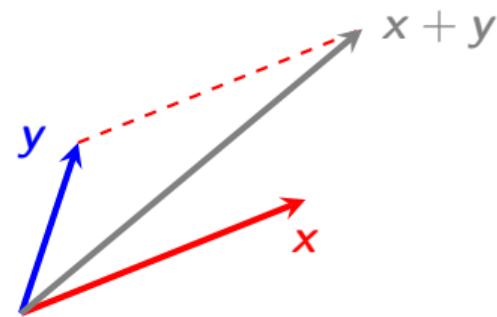
$$2\mathbf{v} = 2 \begin{bmatrix} 3 \\ 1 \end{bmatrix} = \begin{bmatrix} 6 \\ 2 \end{bmatrix}$$



Addition of Vectors

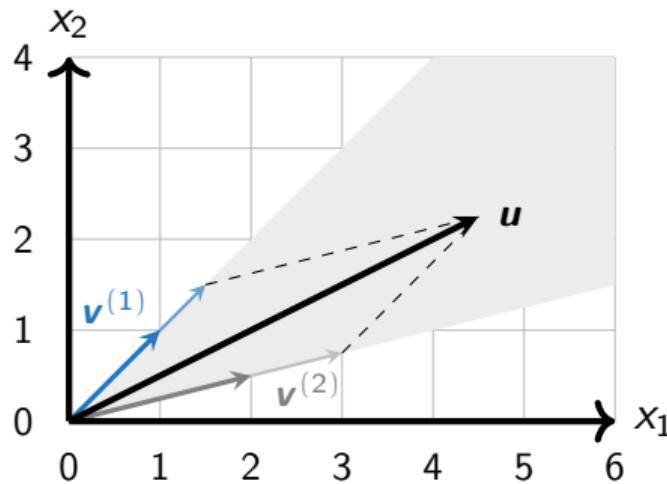
$$\mathbf{x} + \mathbf{y} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} x_1 + y_1 \\ x_2 + y_2 \end{bmatrix}$$

$$\mathbf{v} + \mathbf{w} = \begin{bmatrix} 3 \\ 1 \end{bmatrix} + \begin{bmatrix} 2 \\ 4 \end{bmatrix} = \begin{bmatrix} 5 \\ 5 \end{bmatrix}$$



Linear Combination of Vectors

$$\mathbf{u} = c_1 \mathbf{v}^{(1)} + c_2 \mathbf{v}^{(2)} + \cdots + c_n \mathbf{v}^{(n)} \quad (1)$$



Vector Transpose, inner and outer Product

- Vector transpose:

$$\mathbf{v} = \begin{bmatrix} 3 \\ 1 \end{bmatrix} \quad \mathbf{v}^T = \begin{bmatrix} 3 & 1 \end{bmatrix}$$

- Inner product / dot product / scalar product:

$$\mathbf{v}^T \mathbf{w} \equiv \langle \mathbf{v}, \mathbf{w} \rangle = \begin{bmatrix} 3 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 4 \end{bmatrix} = \sum_{j=1}^m v_j w_j = 3 \cdot 2 + 1 \cdot 4 \quad (2)$$

Vector Transpose and inner and outer Product (Ctd.)

- Outer product:

$$\mathbf{v}\mathbf{w}^T = \begin{bmatrix} 3 \\ 1 \end{bmatrix} \begin{bmatrix} 2 & 4 \end{bmatrix} = \begin{bmatrix} 6 & 12 \\ 2 & 4 \end{bmatrix} \quad (3)$$

The inner product yields a scalar value, the results of an outer product is a matrix!

Length of a Vector

- Length of a vector (Frobenius norm):

$$\|x\| = \sqrt{x^T x} \quad (4)$$

$$\|cx\| = |c| \cdot \|x\| \quad (5)$$

$$\|x + y\| \leq \|x\| + \|y\| \quad (6)$$

- Example:

$$\|\mathbf{v}\| = \sqrt{3^2 + 1^2} = \sqrt{10}$$

Angle between Vectors

- The angle between two vectors is given by:

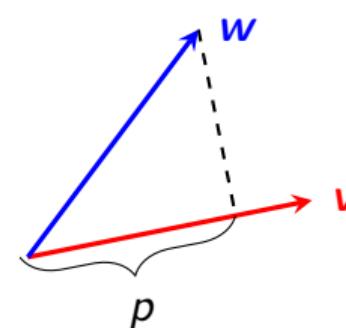
$$\cos \angle(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \cdot \|\mathbf{y}\|} = \frac{\sum_{j=1}^m x_j \cdot y_j}{\sqrt{\sum_{j=1}^m (x_j)^2} \cdot \sqrt{\sum_{j=1}^m (y_j)^2}} \quad (7)$$

$$\cos \angle(\mathbf{v}, \mathbf{w}) = \frac{\mathbf{v} \cdot \mathbf{w}}{\|\mathbf{v}\| \cdot \|\mathbf{w}\|} = \frac{10}{\sqrt{10} \cdot \sqrt{20}} \approx 0.71$$

- Inner product: $\mathbf{x} \cdot \mathbf{y} = \|\mathbf{x}\| \cdot \|\mathbf{y}\| \cdot \cos \angle(\mathbf{x}, \mathbf{y})$

Projection of Vectors

- How is the projection of w onto v defined?
- Formally, we have:

$$\begin{aligned} p &= \|w\| \cos \angle(v, w) \\ &= \|w\| \frac{v \cdot w}{\|v\| \cdot \|w\|} \\ &= \frac{v \cdot w}{\|v\|} \end{aligned} \tag{8}$$


- Note that p is **not a vector!**

What is a Matrix?

General case ($\mathbb{R}^{n \times m}$):

$$\mathbf{X} = \begin{bmatrix} X_{11} & X_{12} & \dots & X_{1m} \\ X_{21} & X_{22} & \dots & X_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ X_{n1} & X_{n2} & \dots & X_{nm} \end{bmatrix}$$

$$\mathbf{M} = \begin{bmatrix} 3 & 4 & 5 \\ 1 & 0 & 1 \end{bmatrix} \quad \mathbb{R}^{2 \times 3}$$

$$\mathbf{N} = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 7 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \mathbb{R}^{3 \times 3}$$

$$\mathbf{P} = \begin{bmatrix} 10 & 1 \\ 11 & 2 \end{bmatrix} \quad \mathbb{R}^{2 \times 2}$$

Matrix Transpose and Addition

- Transpose of a matrix:

$$\mathbf{M}^T = \begin{bmatrix} 3 & 4 & 5 \\ 1 & 0 & 1 \end{bmatrix}^T = \begin{bmatrix} 3 & 1 \\ 4 & 0 \\ 5 & 1 \end{bmatrix}$$

- Addition of matrices:

$$\mathbf{X} + \mathbf{Y} = \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix} + \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix} = \begin{bmatrix} X_{11} + Y_{11} & X_{12} + Y_{12} \\ X_{21} + Y_{21} & X_{22} + Y_{22} \end{bmatrix} \quad (9)$$

Matrix Multiplication

- Multiplication by scalars:

$$c\mathbf{X} = c \begin{bmatrix} X_{11} & X_{12} & X_{13} \\ X_{21} & X_{22} & X_{23} \end{bmatrix} = \begin{bmatrix} c \cdot X_{11} & c \cdot X_{12} & c \cdot X_{13} \\ c \cdot X_{21} & c \cdot X_{22} & c \cdot X_{23} \end{bmatrix} \quad (10)$$

- Matrix-vector multiplication:

$$\mathbf{z} = \mathbf{X}\mathbf{y} = \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} X_{11} \cdot y_1 + X_{12} \cdot y_2 \\ X_{21} \cdot y_1 + X_{22} \cdot y_2 \end{bmatrix} \quad (11)$$

Matrix Multiplication (Ctd.)

- Matrix-matrix multiplication:

$$Z = XY$$

$$\begin{aligned} &= \begin{bmatrix} X_{11} & X_{12} & X_{13} \\ X_{21} & X_{22} & X_{23} \end{bmatrix} \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \\ Y_{31} & Y_{32} \end{bmatrix} \\ &= \begin{bmatrix} X_{11}Y_{11} + X_{12}Y_{21} + X_{13}Y_{31} & X_{11}Y_{12} + X_{12}Y_{22} + X_{13}Y_{32} \\ X_{21}Y_{11} + X_{22}Y_{21} + X_{23}Y_{31} & X_{21}Y_{12} + X_{22}Y_{22} + X_{23}Y_{32} \end{bmatrix} \quad (12) \end{aligned}$$

Matrix Inversion

- Matrix inversion is defined for **square matrices** $X \in \mathbb{R}^{n \times n}$
- A matrix X multiplied by its inverse X^{-1} gives the **identity matrix**:

$$X^{-1}X = XX^{-1} = I \quad (13)$$

$$I = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix}$$

- If X^{-1} exists, we say that X is **non-singular**

Matrix Inversion (Ctd.)

- It holds that (C is the **cofactor matrix**):

$$\mathbf{X}^{-1} = \frac{1}{\det(\mathbf{X})} \mathbf{C}^T \quad (14)$$

- A condition for invertability is that **the determinant has to be different from zero**
- Example:**

$$\mathbf{X} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \quad \det(\mathbf{X}) = 0 \quad \mathbf{X}^{-1} = ?$$

Matrix Inversion Example

$$X = \begin{bmatrix} 1 & 1/2 \\ -1 & 1 \end{bmatrix} \quad X^{-1} = \begin{bmatrix} 2/3 & -1/3 \\ 2/3 & 2/3 \end{bmatrix}$$

Please verify!

$$XX^{-1} = I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = X^{-1}X$$

Use for example the Gauss-Jordan algorithm to find the inverse!

Matrix Pseudoinverse

- **Question:** How can we invert a matrix $X \in \mathbb{R}^{n \times m}$ which is not squared?
- **Left pseudoinverse** $X^\# X$:

$$X^\# X = \underbrace{(X^\top X)^{-1} X^\top}_{\text{left-multiplied}} X = I_m \quad (15)$$

- **Right pseudoinverse** $XX^\#$:

$$XX^\# = X X^\top \underbrace{(X X^\top)^{-1}}_{\text{right-multiplied}} = I_n \quad (16)$$

Eigenvectors and Eigenvalues

- Some vectors v only change their length when multiplied by a matrix X
- Example:

$$\begin{bmatrix} 4 & -1 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} = 2 \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

- These vectors are called **eigenvectors** of X , the scaling factors are known as **eigenvalues** of X
- More general:

$$Xv = \lambda v \tag{17}$$

Eigenvectors form a Basis (Eigenbasis)

- Let us assume that there are n eigenvectors with corresponding eigenvalues:

$$\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \dots, \mathbf{v}^{(n)}$$

$$\lambda_1, \lambda_2, \dots, \lambda_n$$

- Theorem:**

- For an $n \times n$ matrix with eigenvectors $\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \dots, \mathbf{v}^{(n)}$, if they correspond to **distinct** eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_n$, then the set $\{\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \dots, \mathbf{v}^{(n)}\}$ is linearly independent
- Hence, any vector can be expressed as a linear combination of eigenvectors:

$$\mathbf{v} = c_1 \mathbf{v}^{(1)} + c_2 \mathbf{v}^{(2)} + \cdots + c_n \mathbf{v}^{(n)}$$

Symmetric Matrices

- A squared $n \times n$ matrix \mathbf{X} is **symmetric**, iff

$$\forall i, j : \quad X_{ij} = X_{ji} \quad (18)$$

$$\mathbf{X} = \mathbf{X}^T \quad (19)$$

- Some properties:
 - The inverse \mathbf{X}^{-1} is also symmetric
 - **Eigen-decomposition:** \mathbf{X} can be decomposed into $\mathbf{X} = \mathbf{Q}\mathbf{D}\mathbf{Q}^T$, where the columns of \mathbf{Q} are the eigenvectors of \mathbf{X} , and \mathbf{D} is a diagonal matrix whose entries are the corresponding eigenvalues

Positive (semi-)definite Matrices

- A squared symmetric matrix $X^{n \times n}$ is **positive definite**, iff for any vector $y \in \mathbb{R}^n$:

$$y^\top X y > 0 \tag{20}$$

- Or **positive semi-definite**, iff $y^\top X y \geq 0$

Such matrices are important in machine learning. For instance, the covariance matrix is always positive semi-definite.

Section: Statistics



Random Variables

- What is a **random variable**?

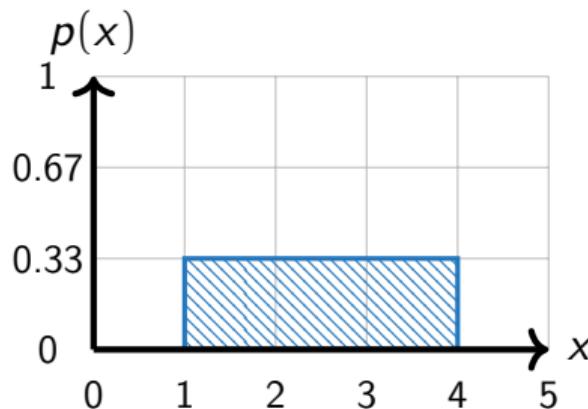
Random Variables

- What is a **random variable**?
 - It's a random number determined by chance (according to a distribution)
 - Random variables in machine learning: input data, output data, noise
- What is a **probability distribution**?

Random Variables

- What is a **random variable**?
 - It's a random number determined by chance (according to a distribution)
 - Random variables in machine learning: input data, output data, noise
- What is a **probability distribution**?
 - Describes the probability that a random variable is equal to a certain value
 - It can be given by the physics of an experiment (e.g. throwing dice)
 - **Discrete** vs. **continuous** distributions

Uniform Distribution



Every outcome is equally probable within a bounded region \mathcal{R}

$$p(x) = 1/\mathcal{R} \quad (21)$$

Discrete Distributions

The random variables take on **discrete values**

Examples:

- When throwing a die, the possible values are given by a countably finite set:

$$x_i \in \{1, 2, 3, 4, 5, 6\}$$

- The number of sand grains at the beach (countably infinite set):

$$x_i \in \mathbb{N}$$

Discrete Distributions (Ctd.)

- All probabilities sum up to 1:

$$\sum_i p(x_i) = 1$$

- Discrete distributions are particularly important in classification
- A discrete distribution is described by a **probability mass function** (also called frequency function)

Bernoulli Distribution

- A **Bernoulli random variable** only takes on two values (e.g. 0 and 1):

$$x \in \{0, 1\} \tag{22}$$

$$p(x = 1|\mu) = \mu \tag{23}$$

$$\text{Bern}(x|\mu) = \mu^x(1 - \mu)^{1-x} \tag{24}$$

$$\mathbb{E}\{x\} = \mu \tag{25}$$

$$\text{var}\{x\} = \mu(1 - \mu) \tag{26}$$

- The only parameter is μ , i.e. the distribution is completely defined by this parameter

Binomial Distribution

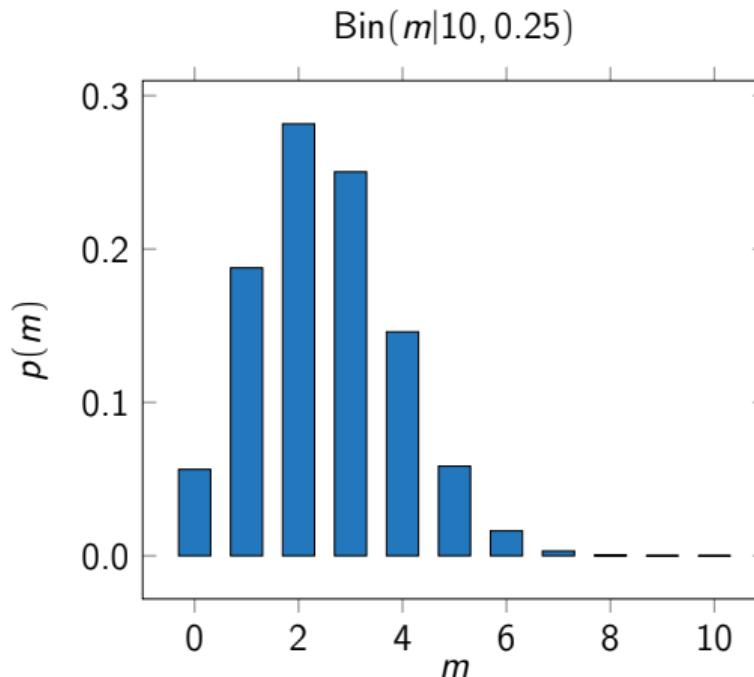
- **Binomial variables** are a sequence of n repeated Bernoulli variables
- **Example:** What is the probability of getting $m \in \mathbb{N}$ heads in n trials?

$$\text{Bin}(m|n, \mu) = \binom{n}{m} \mu^m (1 - \mu)^{n-m} \quad (27)$$

$$\mathbb{E}\{m\} = n\mu \quad (28)$$

$$\text{var}\{m\} = n\mu(1 - \mu) \quad (29)$$

Binomial Distribution (Ctd.)



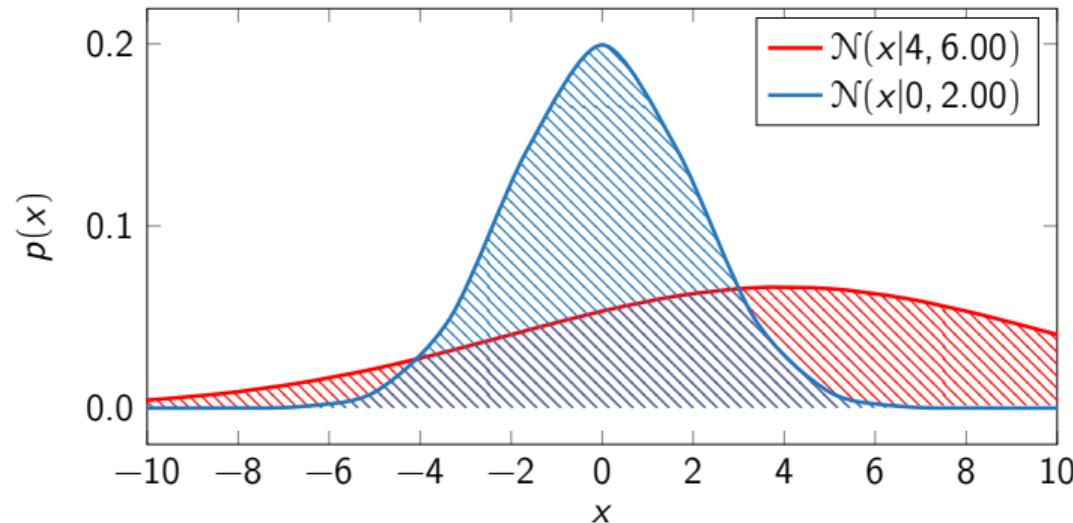
Continuous Distributions

The random variables take on **continuous values**

- Continuous distributions are discrete distributions where the **number of discrete values goes to infinity**, while the **probability of each value goes to zero**
- It's described by a **probability density function** which integrates to 1:

$$\int_{-\infty}^{+\infty} p(x) \, dx = 1$$

Gaussian Distribution



$$p(x) = \mathcal{N}(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left\{ -\frac{(x-\mu)^2}{2\sigma^2} \right\} \quad (30)$$

Central Limit Theorem

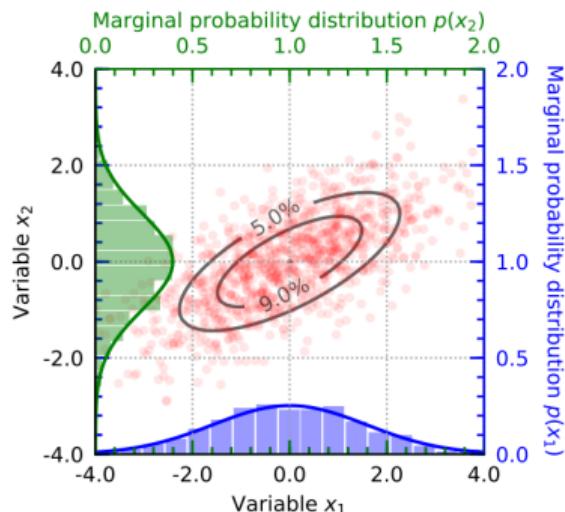
Central Limit Theorem:

The distribution of the sum of n i.i.d. (independent and identically distributed) random variables becomes increasingly Gaussian as n increases

- The Gaussian distribution is one among the most important distributions
- Gaussians are often a good model
- Working with Gaussians leads to **analytical solutions for complex operations**

Multivariate Gaussian Distribution

$$p_D(\mathbf{x}) = \mathcal{N}_D(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^D |\boldsymbol{\Sigma}|}} \exp \left\{ -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \right\} \quad (31)$$



For clarification: \mathbf{x} and $\boldsymbol{\mu}$ are vectors, while $\boldsymbol{\Sigma}$ is a matrix. The probability given by $\mathcal{N}_D(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \in [0; 1]$ is still a scalar value!

Basic Rules of Probability

- Joint distribution:

$$p(x, y) \tag{32}$$

- Marginal distribution:

$$p(y) = \int_x p(x, y) dx \tag{33}$$

- Conditional distribution:

$$p(y|x) = \frac{p(x, y)}{p(x)} \tag{34}$$

Basic Rules of Probability (Ctd.)

- Probabilistic independence:

$$p(x, y) = p(x)p(y) \quad (35)$$

- Chain rule of probabilities:

$$\begin{aligned} p(x_1, \dots, x_n) &= p(x_1|x_2, \dots, x_n)p(x_2, \dots, x_n) \\ &= p(x_1|x_2, \dots, x_n)p(x_2|x_3, \dots, x_n) \dots p(x_{n-1}|x_n)p(x_n) \end{aligned} \quad (36)$$

- Bayes' rule:

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)} \quad (37)$$

Expectation

$$\mathbb{E}_{x \sim p(x)}\{f(x)\} = \mathbb{E}_x\{f\} = \mathbb{E}\{f\} = \sum_x p(x)f(x) \quad \text{discrete case} \quad (38)$$

$$= \int_x p(x)f(x) \, dx \quad \text{continuous case} \quad (39)$$

Approximate expectation:

$$\mathbb{E}\{f\} = \int_x p(x)f(x) \, dx \approx \frac{1}{n} \sum_{i=1}^n f(x_i) \quad (40)$$

Expectation (Ctd.)

- Some rules of expectations:
 - $\mathbb{E}\{a\mathbf{x}\} = a\mathbb{E}\{\mathbf{x}\}$
 - $\mathbb{E}\{\mathbf{x} + \mathbf{y}\} = \mathbb{E}\{\mathbf{x}\} + \mathbb{E}\{\mathbf{y}\}$
 - $\mathbb{E}\{\mathbf{x}\mathbf{y}\} = \mathbb{E}\{\mathbf{x}\}\mathbb{E}\{\mathbf{y}\}$ (if \mathbf{x} and \mathbf{y} are independent)
 - $\mathbb{E}\{\sum_i a_i x_i\} = \sum_i a_i \mathbb{E}\{x_i\}$
- Expectations of functions:
 - $\mathbb{E}\{g(\mathbf{x})\} = \int_{\mathbf{x}} p(\mathbf{x})g(\mathbf{x}) d\mathbf{x}$
 - In general: $\mathbb{E}\{g(\mathbf{x})\} \neq g(\mathbb{E}\{\mathbf{x}\})$

Variance and Covariance

- Covariances give a measure of correlation: (how much variables change together)
- Scalars:

$$\begin{aligned}\text{cov}\{x, y\} &= \mathbb{E}_{x,y}\{(x - \mathbb{E}_x\{x\})(y - \mathbb{E}_y\{y\})\} \\ &= \mathbb{E}_{x,y}\{xy\} - \mathbb{E}_x\{x\}\mathbb{E}_y\{y\}\end{aligned}\tag{41}$$

- Vector notation:

$$\text{cov}\{\mathbf{x}, \mathbf{y}\} = \mathbb{E}_{\mathbf{x},\mathbf{y}}\{(\mathbf{x} - \mathbb{E}_{\mathbf{x}}\{\mathbf{x}\})(\mathbf{y} - \mathbb{E}_{\mathbf{y}}\{\mathbf{y}\})^T\}\tag{42}$$

Kullback-Leibler Divergence

- The **Kullback-Leibler (KL) divergence** is a similarity measure between two distributions p and q :

$$\text{KL}(p\|q) = \sum_x p(x) \cdot \log \frac{p(x)}{q(x)} \quad (43)$$

- Some properties:
 - It is not a distance metric: $\text{KL}(p\|q) \neq \text{KL}(q\|p)$
 - It is non-negative: $\text{KL}(p\|q) \geq 0$
 - If $\forall x : p(x) = q(x) \Rightarrow \text{KL}(p\|q) = 0$

Section:
Optimization



Motivation

- In every machine learning problem, you will have:
 - ① An **objective function** you want to optimize
 - ② **Data** you want to learn from
 - ③ **Parameters** which need to be learned
 - ④ Assumptions about the problem and the data
- We would like to have general solutions to the problem of learning
- Different algorithms embody different objective functions and assumptions

Every machine learning problem is an optimization problem!

Unconstrained Optimization

You know how to do that, don't you?

Constrained Optimization

Formalization:

$$\min_{\theta} \mathcal{J}(\theta) = \dots \quad \leftarrow \text{cost function / objective}$$

$$\text{s. t. } f(\theta) = 0 \quad \leftarrow \text{equality constraints}$$

$$g(\theta) \geq 0 \quad \leftarrow \text{inequality constraints}$$

What should an ideal optimization problem, i. e. the cost function as well as the constraints, look like?

Constrained Optimization (Ctd.)

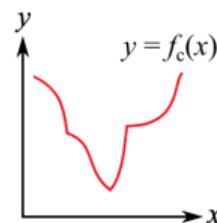
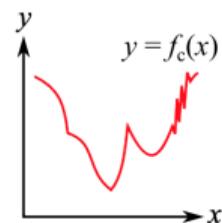
$$\min_{\theta} \mathcal{J}(\theta) = \dots \quad \leftarrow \text{convex function}$$

$$\text{s. t. } f(\theta) = 0 \quad \leftarrow \text{linear function}$$

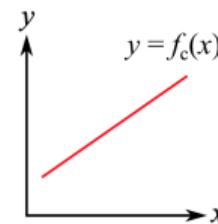
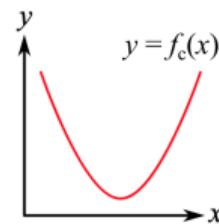
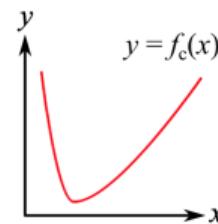
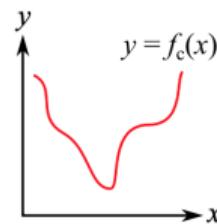
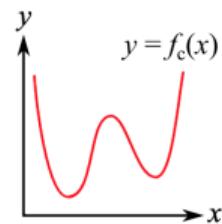
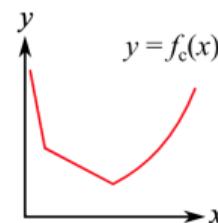
$$g(\theta) \geq 0 \quad \leftarrow \text{convex set}$$

Cost Functions

non-convex



convex

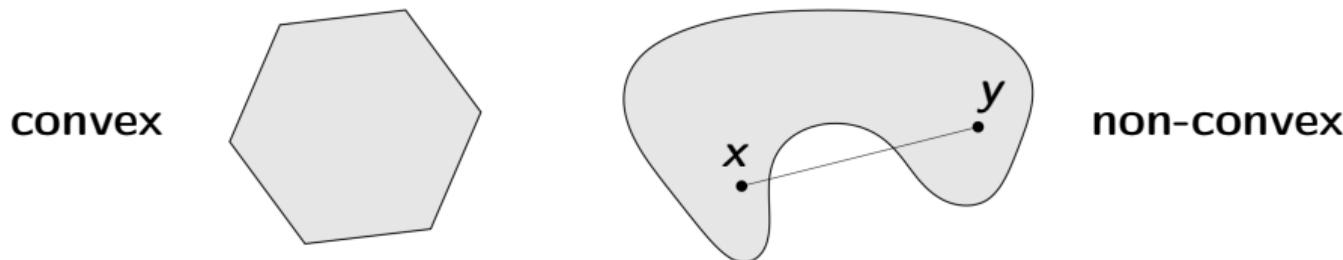


Convexity – Convex Sets

- A set $C \subseteq \mathbb{R}^n$ is convex, if $\forall x, y \in C$ and $\forall \alpha \in [0, 1]$:

$$\alpha x + (1 - \alpha)y \in C \quad (44)$$

- This is the equation of the line segment between x and y

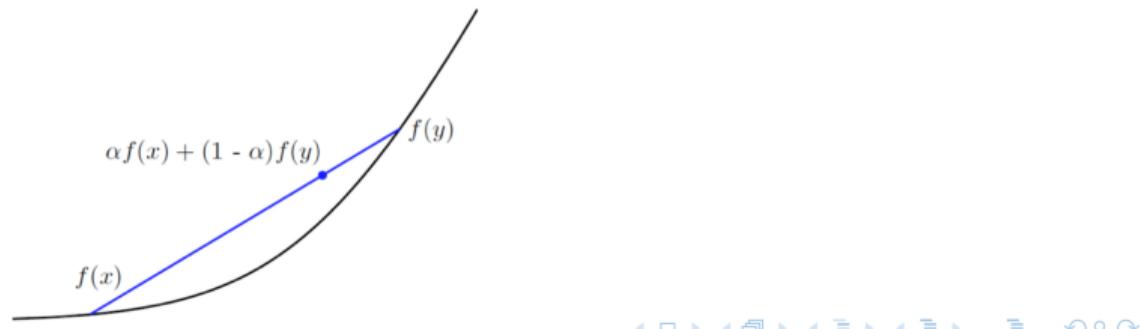


Convexity – Convex Functions

- A function $f : \mathbb{R}^n \mapsto \mathbb{R}$ is convex, if $\forall x, y \in \text{dom}(f)$ and $\forall \alpha \in [0, 1]$:

$$f(\alpha \mathbf{x} + (1 - \alpha) \mathbf{y}) \leq \alpha f(\mathbf{x}) + (1 - \alpha) f(\mathbf{y}) \quad (45)$$

- Examples are linear functions $f(\mathbf{x}) = \mathbf{a}^\top \mathbf{x} + b$ and quadratic functions $f(\mathbf{x}) = \mathbf{x}^\top \mathbf{A}\mathbf{x} + \mathbf{b}^\top \mathbf{x} + c$



Convexity (Ctd.)

- Why are convex cost functions so appealing?
- Local solutions are global optima
- Efficient implementations of optimizers are available

Constrained Optimization

- How to solve this optimization problem?

$$\min_{x,y} \mathcal{J}(x, y) = 2y + x$$

subject to (s.t.):

$$f(x, y) = y^2 + xy - 1 = 0$$

- Convert the problem to an unconstrained one
- This is done using **Lagrange multipliers** λ

The Concept of Lagrange Multipliers

General Lagrange function: $\mathcal{L}(x, y, \lambda) = \mathcal{J}(x, y) + \lambda f(x, y)$

Step ❶: Differentiate w. r. t. x , y and λ :

$$\min_{x,y} \mathcal{J}(x, y) = 2y + x$$

I. $\nabla_x \mathcal{L} = 1 + \lambda y$

s. t.:

$$f(x, y) = y^2 + xy - 1 = 0$$

II. $\nabla_y \mathcal{L} = 2 + 2\lambda y + \lambda x$

III. $\nabla_\lambda \mathcal{L} = y^2 + xy - 1$

The Concept of Lagrange Multipliers (Ctd.)

Step ②: Set equations to zero:

$$\text{I. } 1 + \lambda y \stackrel{!}{=} 0$$

$$\text{II. } 2 + 2\lambda y + \lambda x \stackrel{!}{=} 0$$

$$\text{III. } y^2 + xy - 1 \stackrel{!}{=} 0$$

Step ③: Substitute:

$$\text{I. } \lambda = -\frac{1}{y}$$

$$\text{I.} \rightarrow \text{II. } x = 0$$

$$\text{II.} \rightarrow \text{III. } y = \pm 1$$

Exercise

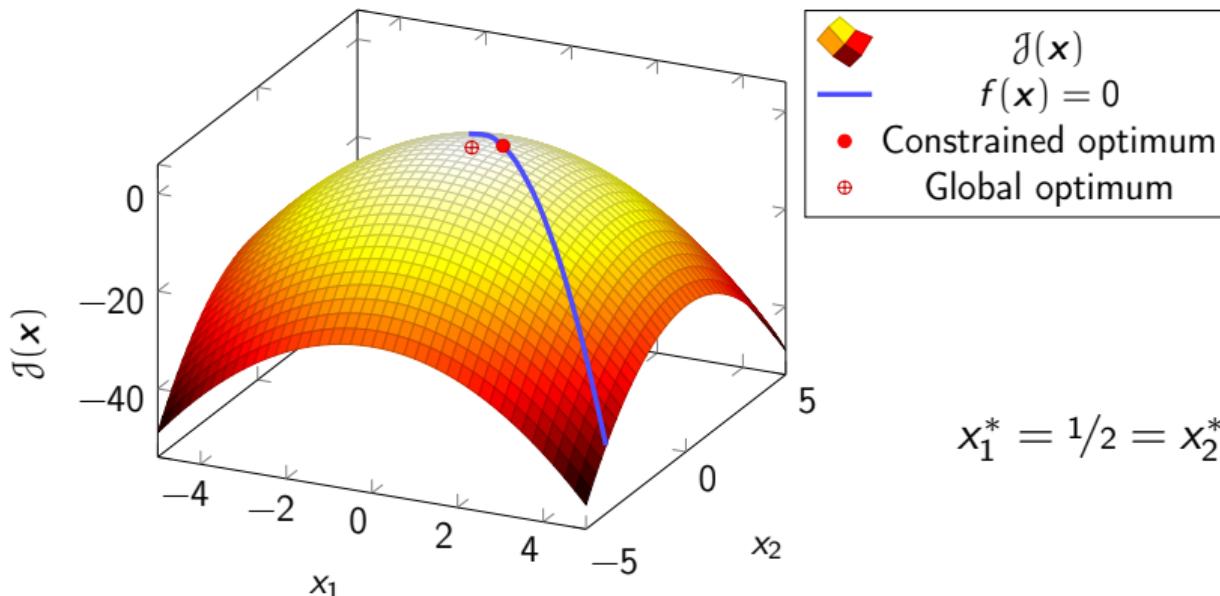
Optimize the following function:

$$\max_{x_1, x_2} \mathcal{J}(x_1, x_2) = 1 - x_1^2 - x_2^2$$

s. t.:

$$f(x_1, x_2) = x_1 + x_2 - 1 = 0$$

Exercise Solution



Numerical Optimization

- Different numerical optimization algorithms exist for optimizing a function numerically on a computer if we can't solve it analytically
- **Gradient descent:** Incrementally update an estimate of the parameters:

$$\boldsymbol{\theta}_{new} \leftarrow \boldsymbol{\theta}_{old} + \alpha \delta \boldsymbol{\theta} \quad (46)$$

- After each update: $\mathcal{J}(\boldsymbol{\theta}_{new}) < \mathcal{J}(\boldsymbol{\theta}_{old})$
- The algorithms differ in the number of iterations required, the computational cost, the convergence guarantees, the robustness with noisy cost functions and their memory usage

Numerical Optimization Algorithms

- **Gradient-based methods:**
 - Gradient descent (with constant or variable step size α)
 - (L-)BFGS (Broyden-Fletcher-Goldfarb-Shanno)
 - Conjugate gradient descent
- **Non-gradient based methods:**
 - Genetic algorithms (e.g. differential evolution)
 - Non-linear simplex
 - Nelder-Mead

Numerical techniques may not find the global optimum!

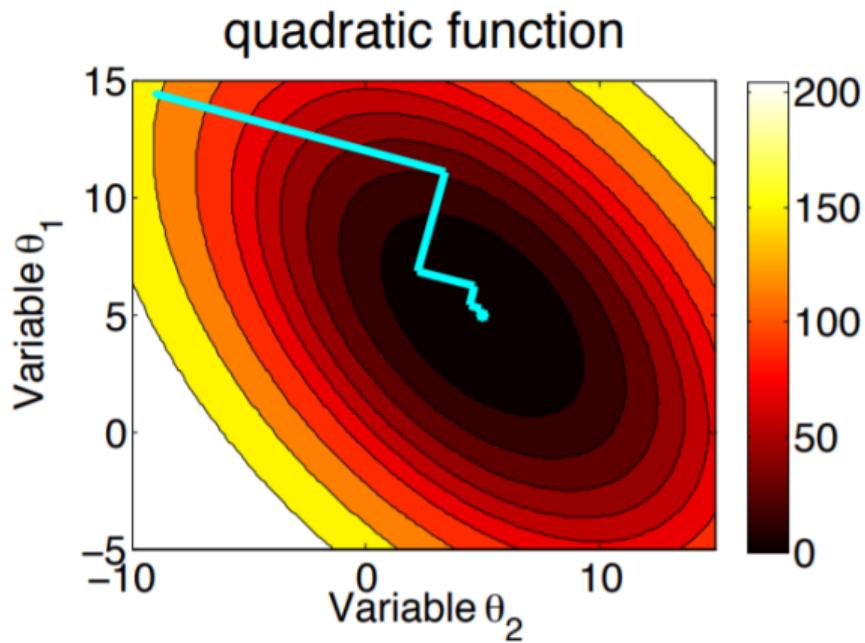
Gradient Descent

- Most basic algorithm (and most commonly used)
- Go into the direction of the **steepest descent**
- The gradient points in the direction of the maximum (\rightarrow subtract gradient)

$$\boldsymbol{\theta}^{(new)} \leftarrow \boldsymbol{\theta}^{(old)} - \alpha \nabla_{\boldsymbol{\theta}} \mathcal{J}(\boldsymbol{\theta}^{(old)}) \quad (47)$$

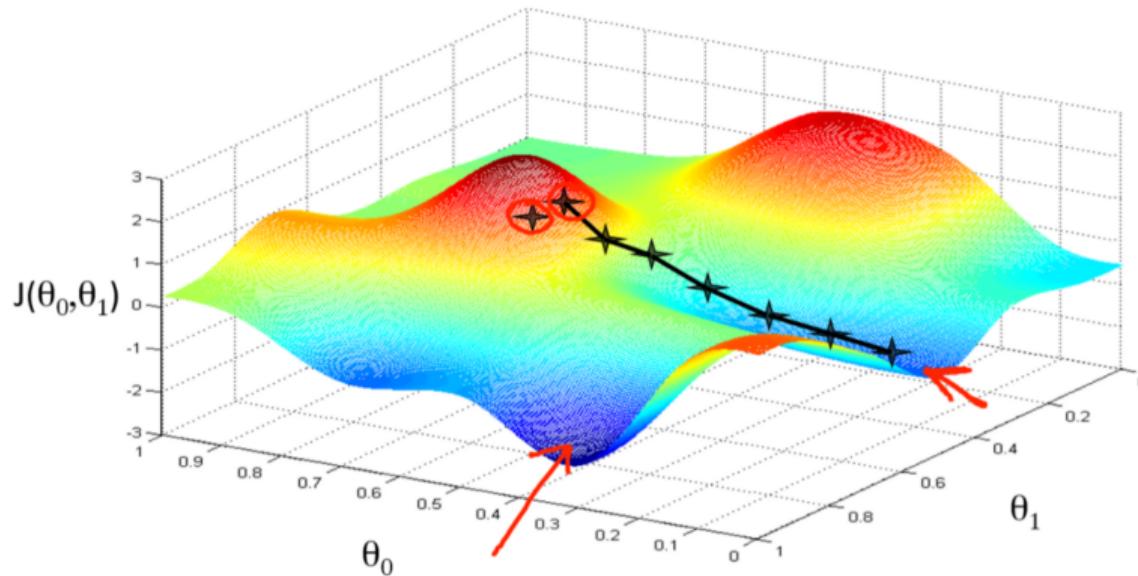
- The parameter updates tend to 'zig-zag' down the valley (cf. next slide)
- Gradient descent is a **1st-order method**

Gradient Descent (Ctd.)



Initialization

Initialization also matters...





Newton's Method

- We want to solve: (H is the **Hessian**, \mathbf{g} the **Jacobian**)

$$\delta\theta = \arg \min_{\delta\theta} \left[c + \mathbf{g}^\top \delta\theta + \frac{1}{2} \delta\theta^\top H \delta\theta \right] \quad (48)$$

Taylor series expansion

- We have to differentiate and set to zero:

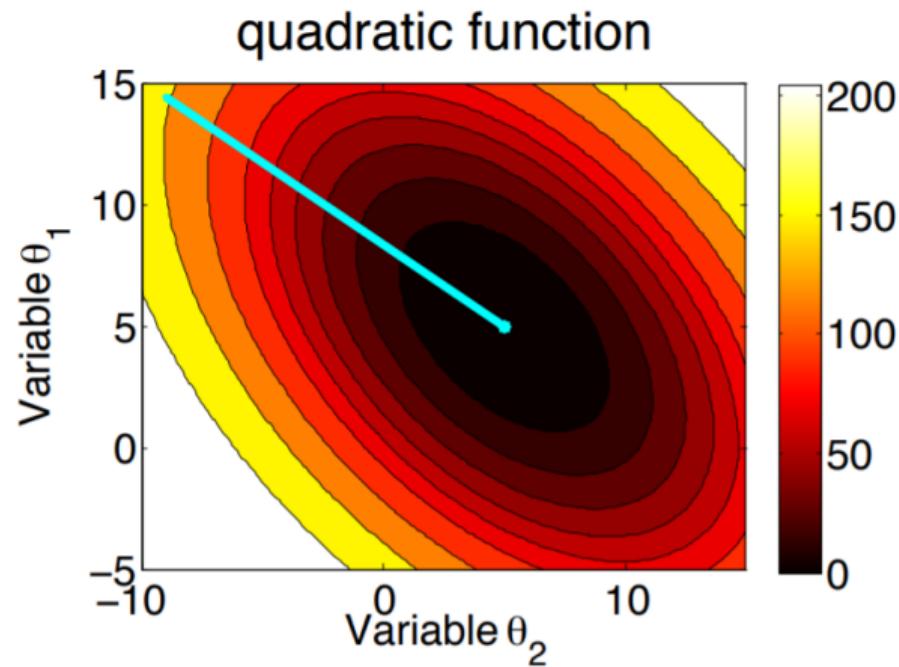
$$\nabla_{\delta\theta} \left[c + \mathbf{g}^\top \delta\theta + \frac{1}{2} \delta\theta^\top H \delta\theta \right] = \mathbf{g} + H \delta\theta \stackrel{!}{=} 0 \quad (49)$$

- Which yields the solution:

$$\delta\theta = -H^{-1}\mathbf{g} \quad (50)$$



Newton's Method (Ctd.)



Section:
Wrap-Up



Summary I

- Machine learning is math!
- Linear algebra:
 - You should know what vectors are and what you can do with them (addition, multiplication, transpose, ...)
 - The same applies to matrices
 - Matrix inversion and pseudoinverse
 - Eigenvectors and eigenvalues are important, **eigenvectors form a basis**

Summary II

- **Statistics:**

- Random variables are numbers **determined by chance**
- Probability distributions describe a **probability mass or density**
- Discrete distributions: Bernoulli, Binomial, Poisson (not covered)
- Continuous distributions: Gaussian, Student-t (not covered)
- Gaussians are important in machine learning and have appealing properties
- Terms: Joint-, marginal- and conditional distribution, chain rule, probabilistic independence, Bayes' rule
- You should know what expectation and variance is

Summary III

- **Optimization:**
 - Every machine learning problem is an optimization problem!
 - Good cost functions are convex
 - Unconstrained and constrained optimization (Lagrange multipliers)
 - Closed-form solutions are not always possible → numerical optimization
 - The most prominent numerical technique is called gradient descent

Self-Test Questions

- ① What is a vector and what is a matrix?
- ② What is the result of an inner product / outer product?
- ③ How can you invert matrices? Is this always possible?
- ④ What is an eigenvalue problem? Where do they play a role?
- ⑤ What are random variables and probability distributions?
- ⑥ Why is the Gaussian distribution so important?
- ⑦ What is Bayes' rule? Explain its components!
- ⑧ What is convexity? Why should cost functions be convex?
- ⑨ What can you do if no analytical solution to optimization problems exists?

What's next...?

- Unit I** Machine Learning Introduction
- Unit II** Mathematical Foundations
- Unit III** **Bayesian Decision Theory**
- Unit IV** Regression
- Unit V** Classification I
- Unit VI** Evaluation
- Unit VII** Classification II
- Unit VIII** Clustering
- Unit IX** Dimensionality Reduction

Thank you very much for the attention!

Topic: *** Applied Machine Learning Fundamentals *** Mathematical Foundations

Term: Winter term 2023/2024

Contact:

Daniel Wehner, M.Sc.

SAP SE / DHBW Mannheim

daniel.wehner@sap.com

Do you have any questions?