

Blatt 1

Das Praktikum kann in **Matlab** ODER in **Python** bearbeitet werden. In beiden Welten gibt es viele Code-Beispiele im Netz. Matlab ist "homogener" zu nutzen.

Für Python wird Linux empfohlen. Bei einer Python-Installation unter Windows wird Anaconda empfohlen.

Alle **Abgaben** erfolgen als gepackte Zip-Dateien per email. Diese sollen neben den Originaldateien auch **html-Exporte** mit den Ergebnissen enthalten.

Aufgabe 1: Einarbeitung

Matlab

Starten Sie Matlab: (In Linux per `matlab &` oder) in Windows per Doppelklick auf das Matlab-Symbol.

Starten Sie rechts oben: `Help` → `Matlab Help`.

Machen Sie sich mit dem Help-Tool vertraut:

- links oben die Registerkarten Contents, Index, Search, Demos;
- Geben Sie unter „search for“ z.B. einmal „exit“ ein;

Arbeiten Sie dann folgende Kapitel unter 'Contents' durch, indem Sie zunächst die Kapitel **kurz überfliegen**, um einen Gesamtüberblick zu bekommen, und vertiefen Sie sie dann, indem Sie die ein oder anderen Befehle selbst im Command-Window ausprobieren ¹. Dafür sollte etwa eine halbe Stunde reichen:

„Matlab → Getting Started →“

- Introduction → Starting and Quitting Matlab;
- Matrices and Arrays; (Matrices and Magic Squares → „The magic function“ auslassen)
- Graphics → Basic Plotting Functions →
 - Creating a Plot,
 - Figure Windows,
 - Saving Figures;
- Graphics → Images;
- Programming →

¹auch hier zu finden: <http://www.mathworks.com/access/helpdesk/help/techdoc/matlab.html>

- Flow Control,
- Other Data Structures → Multidimensional Arrays,
- Other Data Structures → Characters and Text,
- Scripts and Functions;

Nun zur Vertiefung noch folgende Kapitel (30 Minuten):

- Matlab → Mathematics → Matrices and Linear Algebra → Function Summary;
- Matlab → Programming → Data Structures →
 - Creating and Concatenating Matrices;
 - Accessing Elements of a Matrix;
 - Getting Information About a Matrix;
 - Resizing and Reshaping Matrices;
 - Multidimensional Arrays;
- Matlab → Programming → Data Types, →
 - Overview;
 - Numeric Types;
 - Logical Types;
- Matlab → Programming → M-File Programming →
 - M-File Scripts and Functions;
 - Function Arguments;
- Matlab → Programming → Programming Tips;

Machen Sie sich noch Folgendes klar:

- Kommentarzeichen
- Unterdrückung der Befehlsausgabe
- Listing aller Variablen einer Sitzung
- Suchpfad für Matlab-Funktionen (m-Files) anzeigen bzw. selbst ergänzen

Python

Machen Sie sich folgende Themen klar:

- Funktionen
- Klassen
- Aufruf eines Jupyter-Notebooks

- Graphikausgabemöglichkeiten (Samples, 3D-Oberfläche für $f(x,y)$)

Folgende Pakete können brauchbar sein:

- **Basis:** numpy, scipy
- **Graphik:** matplotlib, plotly, pandas, seaborn
- **Editoren:** jupyter, PyCharm
- **Mustererkennung:** sklearn, scikit
- u.v.m

Bearbeiten Sie nachfolgend Programmieraufgaben in Matlab- oder Jupyter-Notebooks.

Aufgabe 2a: Statistik

1. Dichte, Erwartungswert, Varianz und Kovarianz

Eine 1D-Zufallsvariable x sei im Intervall $[a, b] \subset \mathbb{R}$ gleichverteilt.

- Skizzieren Sie die Dichte $p(x)$.
Welchen Wert hat die Dichte im Intervall $[a, b]$?
- Berechnen Sie den Erwartungswert:

$$\mu_x = E[x] = \int_a^b xp(x)dx$$

- Berechnen Sie die Varianz:

$$\sigma_x^2 = E[(x - \mu_x)^2] = \int_a^b (x - \mu_x)^2 p(x) dx$$

2. Sample Mean und Kovarianzmatrix

Gegeben ist eine Stichprobe \mathcal{X} einer vektoriellen Zufallsvariable $\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \in \mathbb{R}^2$, $i = 1, \dots, 4$:

$$\mathcal{X} = \left\{ \mathbf{x}_1 = \begin{pmatrix} 1 \\ 2 \end{pmatrix}, \mathbf{x}_2 = \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \mathbf{x}_3 = \begin{pmatrix} -5 \\ 1 \end{pmatrix}, \mathbf{x}_4 = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \right\}$$

- Berechnen Sie den Mittelwertsvektor $\boldsymbol{\mu}_x = \frac{1}{n} \sum_i \mathbf{x}_i$.
- Berechnen Sie die Autokovarianzmatrix $\boldsymbol{\Sigma}_{xx} = \frac{1}{n} \sum_i (\mathbf{x}_i - \boldsymbol{\mu}_x)(\mathbf{x}_i - \boldsymbol{\mu}_x)^\top$.

3. Multivariate Normalverteilung

Eine Zufallsvariable $\mathbf{u} = \begin{pmatrix} x \\ y \end{pmatrix}$ sei normalverteilt mit Mittelwertsvektor $\boldsymbol{\mu}_u = (0, 0)^\top$ und Kovarianzmatrix $\boldsymbol{\Sigma}_{uu} = \begin{pmatrix} 4 & 0 \\ 0 & 1 \end{pmatrix}$.

Eine zweite Zufallsvariable $\mathbf{v} = \begin{pmatrix} x \\ y \end{pmatrix}$ sei normalverteilt mit Mittelwertsvektor $\boldsymbol{\mu}_v = (4, 0)^\top$ und Kovarianzmatrix $\boldsymbol{\Sigma}_{vv} = \begin{pmatrix} 1 & 0 \\ 0 & 4 \end{pmatrix}$.

- Skizzieren Sie die Lage und die Ausrichtung der beiden Verteilungen im x/y -Koordinatensystem durch Ellipsen.
- Wie etwa würde die Trennfunktion eines Klassifikators aussehen? (Die A-priori-Wahrscheinlichkeiten der beiden Verteilungen seien gleich, Skizze).

4. Entropie einer Diskreten Verteilung

Es ist folgende diskrete Wahrscheinlichkeitsverteilung gegeben:

$$P(x) = \begin{cases} P_1 = 0.2 & \text{für } x = 1 \\ P_2 = 0.8 & \text{für } x = 2 \end{cases}$$

Verwenden Sie bei den Berechnungen den natürlichen Logarithmus $\ln(x)$.

- Welchen Wert hat die Information $x = 1$?
- Welchen Wert hat die Information $x = 2$?
- Welche Entropie hat diese Verteilung?
- Wenn $P_1 = 0.1$ wäre, wäre die Entropie dann größer oder kleiner? Warum ist das so?

Aufgabe 2b: Statistik

Führen Sie Folgendes in einem Jupyter-Notebook aus und exportieren Sie es als pdf:

- Berechnung von Mittelwertsvektor $\boldsymbol{\mu}$ und Kovarianzmatrix $\boldsymbol{\Sigma}$ für obige Daten
- Plotten von Normalverteilungen für $d=2$ Dimensionen für verschiedene $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$ als
 - 3D-Oberfläche,
 - Höhenlinienbild (Konturbild),
 - 2D-Verteilung von zufälligen Samples