



Mathematisch-algorithmische Grundlagen für Data Science

Aufgabenblatt 1

Abgabe zu **zweit** vor der Vorlesung am 8. April 2025.

Sollpunktzahl: 15 Punkte

Aufgabe 1 (Lineare Unabhängigkeit)

2+2=4 Punkte

(a) Zeigen Sie, dass die Vektoren

$$\mathbf{v}_1 = \begin{pmatrix} -1 \\ 2 \\ 3 \end{pmatrix}, \quad \mathbf{v}_2 = \begin{pmatrix} 3 \\ -4 \\ 1 \end{pmatrix}, \quad \mathbf{v}_3 = \begin{pmatrix} 2 \\ 4 \\ -1 \end{pmatrix}$$

linear unabhängig sind.

(b) Zeigen Sie, dass die Vektoren

$$\mathbf{v}_1 = \begin{pmatrix} -2 \\ 1 \\ 1 \end{pmatrix}, \quad \mathbf{v}_2 = \begin{pmatrix} 1 \\ -2 \\ 1 \end{pmatrix}, \quad \mathbf{v}_3 = \begin{pmatrix} 1 \\ 1 \\ -2 \end{pmatrix}$$

linear abhängig sind.

Aufgabe 2 (Lösbarkeit von linearen Gleichungssystemen)

2+2+2=6 Punkte

Bestimmen Sie, ob die folgenden linearen Gleichungssysteme keine, genau eine oder unendlich viele Lösungen besitzen.

(a)

$$\begin{aligned} 2x_1 - 3x_2 &= 11 \\ 5x_1 - x_2 &= 8 \\ x_1 - 5x_2 &= 16 \end{aligned}$$

(b)

$$\begin{aligned} x_1 + x_2 - x_3 &= 2 \\ -2x_1 \quad \quad \quad x_3 &= -2 \\ 5x_1 - x_2 + 2x_3 &= 4 \\ 2x_1 + 6x_2 - 3x_3 &= 5 \end{aligned}$$

(c)

$$\begin{aligned} x_1 + 2x_2 + 3x_3 + 4x_4 &= 1 \\ -2x_1 - 4x_2 - 6x_3 - 8x_4 &= -2 \\ 3x_1 + 6x_2 + 9x_3 + 12x_4 &= 2 \end{aligned}$$

Aufgabe 3 (Berechnung der Determinante)

4 Punkte

Berechnen Sie für die Matrix

$$\mathbf{A} = \begin{pmatrix} 1 & -1 & -1 & 1 \\ 1 & -2 & -3 & 1 \\ -2 & 6 & 9 & 0 \\ 4 & -3 & -2 & 5 \end{pmatrix}$$

die Determinante $\det(\mathbf{A})$ mithilfe der Leibniz-Formel.**Aufgabe 4 (Cramersche Regel)**

3 Punkte

Lösen Sie das folgende lineare Gleichungssystem mithilfe der Cramerschen Regel (Satz 1.26):

$$\begin{aligned} -2x_1 + 3x_2 + 3x_3 &= 4 \\ 5x_1 - 2x_2 - 4x_3 &= 2 \\ x_1 + x_2 - x_3 &= -2 \end{aligned}$$

Aufgabe 5 (Leibniz-Formel und Cramersche Regel in Python)

4+2=6 Punkte

- (a) Implementieren Sie die Leibniz-Formel zur Berechnung der Determinate einer quadratischen Matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$.

Hinweis: Mit der Funktion `itertools.permutations(range(n))` können Sie alle Permutationen für die Menge $\{0, 1, \dots, n-1\}$ erzeugen.

- (b) Implementieren Sie die Cramersche Regel zur Lösung eines linearen Gleichungssystems $\mathbf{Ax} = \mathbf{b}$ mit einer regulären Matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ in Python.

Hinweis: Nutzen Sie Ihre Funktion aus Teil (a).