01142 Algorithmische Mathematik

KE 1

„Definition": abzählbar

$\mathbb{Q}$ ist abzählbar, $\mathbb{R}$ hingegen nicht.

Wikipedia:

Menge A abzählbar unendlich $\Leftrightarrow |A| = |\mathbb{N}|$, also existiert eine Bijektion $f: A \to \mathbb{N}$

höchstens abzählbare Mengen: abzählbar unendliche Mengen + endliche Mengen

Notation

- $\mathbb{N}$ wird in diesem Kurs inklusive der Null definiert.

Definition: Gaußklammern

Sei $x \in \mathbb{R}$. Es gilt

$\lfloor x \rfloor = \max\{k \in \mathbb{Z} \mid k \leq x\}$, also die nächstkleinere ~~ganze~~ Zahl.

$\lceil x \rceil = \min\{k \in \mathbb{Z} \mid k \geq x\}$, also die nächstgrößere ganze Zahl.

Wir nennen $\lfloor\ \rfloor$ die untere und $\lceil\ \rceil$ die obere Gaußklammer.

Summations- und Produktzeichen

Wir definieren als leere Summe

$\sum_{i=1}^{0} x = 0$ und als leeres Produkt $\prod_{i=1}^{0} x = 1$.

Definition: Mengenfamilie

Wir bezeichnen die Menge M als Mengenfamilie, wenn die Elemente von M selbst Mengen sind.

Vereinigungs- und Schnittmenge

$M \cup N = \{x \mid x \in M \vee x \in N\}$

$M \cap N = \{x \mid x \in M \wedge x \in N\}$

$M \dot{\cup} N = \{x \mid (x \in M \vee x \in N) \wedge M \cap N = \emptyset\}$

Ist $M \neq \emptyset \neq N$, so sagen wir, dass M und M die Menge $M \cup N$ partitionieren, und bezeichnen M, N als Klassen von $M \cup N$.

Differenzmenge

$M \setminus N = \{x \mid x \in M \wedge x \notin N\}$.

Betrachten wir die Menge $M \subseteq X$ bezüglich einer Grundmenge X („Universum"), so nennen wir $\overline{M} = X \setminus M$ das Komplement von M.

Kartesisches Produkt

$M \times N = \{(m,n) \mid m \in M, n \in N\}$.

Potenzmenge

Sei X eine Menge. Wir bezeichnen $2^X = \{Y \mid Y \subseteq X\}$ als Potenzmenge von X, also die Menge aller Teilmengen von X.

Abbildungen

Definition: Permutation

Wir nennen die bijektive Abbildung $\sigma: M \to M$ eine Permutation.

Definition 1.3.1: gleichmächtig

Wir nennen die Mengen A, B gleichmächtig, wenn eine bijektive Abbildung $f: A \to B$ existiert.

Definition: Einschränkung

Sei $f: M \to N$ eine Abbildung und sei $L \subseteq M$. Wir nennen $f|_L : L \to N$ mit $f|_L(x) = f(x)$ eine Einschränkung von f auf L.

Proposition 1.3.2

a.) Die Komposition injektiver Abbildungen ist injektiv.

b.) Die Komposition surjektiver Abbildungen ist surjektiv.

c.) Die Komposition bijektiver Abbildungen ist bijektiv.

„Definition": Geradenarrangement

Ein Geradenarrangement ist eine endliche Menge paarweiser verschiedener Geraden in der Ebene. Die Schnittpunkte der Geraden nennen wir Ecken des Arrangements. Die zusammenhängenden Gebiete, in welche die Ebene durch die Geraden zerteilt wird, nennen wir beschränkte bzw. unbeschränkte Zellen. Zwei Geraden schneiden sich in genau einer Ecke oder sie sind parallel.

„Definition": konvexes n-Eck

Von n Strecken berandetes Gebilde der Ebene, von denen jede genau zwei weitere Strecken in je einer Ecke in einem Winkel $\neq 180°$ trifft. Die Verbindungsstrecke zwischen zwei inneren Punkten zweier verschiedener Strecken trifft keine weitere Strecke.

Elementare Abzählprobleme und diskrete Wahrscheinlichkeiten

Urnenexperiment mit zurücklegen: Variation mit Wiederholung

Proposition 2.1.2

Seien $m, n \in \mathbb{N}$, $m \geq 1$ und sei A eine n-elementige Menge und R eine m-elementige Menge. Dann ist die Anzahl aller Abbildungen $f: A \to R$ gleich m^n bzw. $|R|^{|A|}$.

Korollar 2.1.3

Sei X eine n-elementige Menge mit $n \in \mathbb{N}$. Dann hat X genau 2^n Teilmengen, also $|2^X| = 2^{|X|}$.

Proposition 2.1.4

Sei $n \geq 1$. Jede n-elementige Menge hat genau 2^{n-1} Teilmengen mit ungeraden vielen Elementen und 2^{n-1} Teilmengen mit gerade vielen Elementen.

Injektive Abbildungen, Permutationen und Fakultät

Urnenexperiment ohne zurücklegen: Variation ohne Wiederholung

Proposition 2.2.1

Seien $m, n \in \mathbb{N}$. Dann gibt es genau $m(m-1)\dots(m-n+1) = \prod_{i=0}^{n-1}(m-i)$ injektive Abbildungen einer n-elementigen Menge A in eine gegebene m-elementige Menge R.

Definition 2.2.2: Fakultät

Sei $n \in \mathbb{N}$. Die Zahl $n! = \prod_{i=1}^{n} i$ nennen wir die Fakultät von n.

Definition: Darstellung einer Permutation

Sei $\sigma: A \to A$ eine Permutation mit $|A| = n$. Durch Abzählen der Elemente von A lässt sich σ als Permutation $\sigma: N \to N$ mit $N := \{1, 2, \dots, n\}$ auffassen. Diese kann als Abbildungsmatrix $\begin{pmatrix} 1 & 2 & \cdots & n \\ \sigma(1) & \sigma(2) & \cdots & \sigma(n) \end{pmatrix}$ oder als n-Tupel $(\sigma(1), \sigma(2), \dots, \sigma(n))$ dargestellt werden.

Definition: Zyklus

Sei N eine Menge. Ein Zyklus ist eine wiederholungsfreie endliche oder unendliche Folge $\langle a_1\ a_2 \cdots a_k \rangle$ von Zahlen in N.

Definition: Fixpunkt einer Permutation

Sei $\sigma: N \to N$ eine Permutation und sei $x \in N$. Wir nennen x einen Fixpunkt von σ, wenn $\sigma(x) = x$ gilt.

Definition: Darstellung von Zyklen als Permutation

Sei N eine Menge und sei $\langle a_1\ a_2 \cdots a_k \rangle$ ein Zyklus von N. Wir können den Zyklus selbst wieder als Permutation $\sigma: N \to N$ mit

$$\sigma(a_i) = \begin{cases} a_{i+1} & \text{, falls } i < k \\ a_1 & \text{, falls } i = k \\ a_i & \text{, falls } a_i \in N \setminus \{a_1\ a_2 \cdots a_k\}. \end{cases}$$

Beispiel für das Vertauschen der Zyklenelemente:

Sei $\langle 4\ 2\ 5 \rangle$ ein Zyklus von $\{1, 2, 3, 4, 5\}$. Permutationsdarstellung:

$4 \to 2 \to 5$, alle anderen Elemente bleiben fix.

Ergebnis $\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 5 & 3 & 2 & 4 \end{pmatrix}$.

Definition: Produkt

Seien σ_1 und σ_2 Permutationen. Wir bezeichnen die Komposition $\sigma_1 \circ \sigma_2$ als Produkt. Das Produkt ist ebenfalls eine Permutation.

Definition: Zerlegung von Permutationen in Zyklen

Sei $\sigma: N \to N$ eine Permutation. Schreiben wir σ als Produkt von Zyklen, so sagen wir, dass wir σ in Zyklen zerlegen.

Proposition 2.2.4

Jede Permutation σ lässt sich (bis auf die Reihenfolge eindeutig) in paarweise disjunkte Zyklen zerlegen.

Definition: signierte Teilmenge

Sei X eine Menge. Wir nennen ein Tupel (C_1, C_2) mit $C_1 \cap C_2 = \emptyset$ und $C_1 \cup C_2 \subseteq X$ eine signierte Teilmenge von X.

Proposition:

Ist X eine endliche Menge, so hat X genau $3^{|X|}$ signierte Teilmengen.

Definition: Transposition

Eine Transposition ist eine Permutation, die nur zwei Zahlen vertauscht und alle anderen fix lässt.

Binomialkoeffizienten

Definition 2.3.1: Binomialkoeffizient

Seien $n, k \in \mathbb{N}$, $n \geq k$. Der Binomialkoeffizient n über k ist definiert als

$$\binom{n}{k} := \frac{n(n-1)(n-2)\cdots(n-k+1)}{1 \cdot 2 \cdots (k-1)k} = \frac{\prod_{i=0}^{k-1}(n-i)}{k!}.$$

Alternativ:

$$\binom{n}{k} := \frac{n!}{k!\,(n-k)!}$$

Für $k > n$ definieren wir $\binom{n}{k} = 0$.

Der Binomialkoeffizient gibt die Anzahl an Möglichkeiten an, aus einer n-elementigen Menge eine k-elementige Teilmenge auszuwählen.

Urnenexperiment: Ziehen ohne Zurücklegen und Reihenfolge ignorieren
$\to$ Kombination ohne Wiederholung

Definition 2.3.2

Sei X eine Menge und $k \in \mathbb{N}$. Wir bezeichnen mit $\binom{X}{k}$ die Menge aller k-elementigen Teilmengen von X.

Proposition 2.3.3

Sei X eine n-elementige Menge und sei $k \in \mathbb{N}$. Dann hat X genau $\binom{n}{k}$ k-elementige Teilmengen. Als Formel:

$$\left|\binom{X}{k}\right| = \binom{|X|}{k}.$$

Korollar 2.3.5

Die Anzahl der Partitionen der Zahl n in k Summanden mit Beachtung der Reihenfolge ist $\binom{n+k-1}{k-1}$.

Proposition 2.3.6

Seien $n, k \in \mathbb{N}$, $n \geq k$. Dann gilt:

a.) $\binom{n}{k} = \binom{n}{n-k}$

b.) Sei zusätzlich $n \geq k \geq 1$. Dann ist $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$.

Satz 2.3.7: Binomischer Satz

Sei $n \in \mathbb{N}$. Dann ist $(1+x)^n = \sum_{k=0}^{n} \binom{n}{k} x^k$.

Korollar 2.3.8

Sei $n \in \mathbb{N}$. Es gilt $(a+b)^n = \sum_{k=0}^{n} \binom{n}{k} a^k b^{n-k}$.

Korollar 2.3.9

$$\binom{n}{0} + \binom{n}{1} + \binom{n}{2} + \cdots \binom{n}{n} = 2^n.$$

Proposition 2.3.10

$$\sum_{i=0}^{n} \binom{n}{i}^2 = \binom{2n}{n}.$$

Definition 2.3.11: Multinomialkoeffizient

Sei $k_1 + \dots + k_m = n$. Der Multinomialkoeffizient ist definiert als

$$\binom{n}{k_1, k_2, \dots, k_m} := \frac{n!}{k_1!\, k_2! \cdots k_m!}.$$

Der Multinomialkoeffizient beschreibt die Anzahl der Möglichkeiten, n Objekte, von denen jeweils k_i nicht unterscheidbar sind, anzuordnen.

Satz 2.3.12: Multinomialsatz

Sei $n \in \mathbb{N}$. Dann ist

$$\sum_{\substack{i=1 \\ k_i \geq 1}}^{m} \binom{n-1}{k_1, \dots, k_{i-1}, k_i - 1, k_{i+1}, \dots, k_m} = \binom{n}{k_1, \dots, k_m},$$

$$(x_1 + x_2 + \dots + x_m)^n = \sum_{\substack{k_1 + \dots + k_m = n \\ k_1, \dots, k_m \geq 0}} \binom{n}{k_1, k_2, \dots, k_m} x_1^{k_1} x_2^{k_2} \cdots x_m^{k_m}.$$

Abschätzungen

Definition 2.4.1: „Big-O-Notation"

Seien $f, g: \mathbb{N} \to \mathbb{R}$ Abbildungen. Wir schreiben $f = O(g)$ oder $f(n) = O(g(n))$, wenn es eine Konstante c und einen Startpunkt $n_1 \in \mathbb{N}$ gibt, so dass stets $|f(n)| \leq c\, g(n)$ für alle $n \in \mathbb{N}$ mit $n \geq n_1$ gilt.

Proposition 2.4.2

Seien $c, a, \alpha, \beta > 0$ feste reelle, positive Zahlen unabhängig von n. Dann gilt:

a.) $\alpha \leq \beta \Rightarrow n^\alpha = O(n^\beta)$

b.) $a > 1 \Rightarrow n^c = O(a^n)$

c.) $\alpha > 0 \Rightarrow (\ln n)^c = O(n^\alpha)$

Definition: Landau-Symbole

- $f(n) = o(g(n)) \Leftrightarrow \lim_{n \to \infty} \frac{f(n)}{g(n)} = 0$; f wächst echt langsamer als g
- $f(n) = \Omega(g(n)) \Leftrightarrow g(n) = O(f(n))$; $g(n)$ ist eine untere Schranke für $f(n)$ für große n.
- $f(n) = \omega(g(n)) \Leftrightarrow g(n) = o(f(n))$; f ist asymptotisch dominant für g, wächst also viel schneller.
- $f(n) = \Theta(g(n)) \Leftrightarrow f(n) = O(g(n))$ und $f(n) = \Omega(g(n))$; f und g verhalten sich bis auf einen konstanten Faktor asymptotisch gleich, genauer $\exists c_1, c_2 > 0,\ n_0 \in \mathbb{N}\ \forall n \geq n_0: c_1 g(n) \leq f(n) \leq c_2 g(n)$
- $f(n) \sim g(n) \Leftrightarrow \lim_{n \to \infty} \frac{f(n)}{g(n)} = 1$

Proposition 2.5.1

$2^{n-1} \leq n! \leq n^n$

Satz 2.5.2

$\forall n \geq 1: n^{\frac{n}{2}} \leq n! \leq \left(\frac{n+1}{2}\right)^n$

Lemma 2.5.3: Ungleichung arithmetischen-geometrischen Mittel

Seien $a, b \in \mathbb{R}^+$. Dann ist $\sqrt{ab} \leq \frac{a+b}{2}$.

Satz 2.5.4

$\forall n \in \mathbb{N} \setminus \{0\}: e\left(\frac{n}{e}\right)^n \leq n! \leq en\left(\frac{n}{e}\right)^n$.

Stirlingsche Formel

$n! \sim \sqrt{2\pi n}\left(\frac{n}{e}\right)^n$.

Satz 2.5.5

Seien $1 \leq k \leq n \in \mathbb{N}$. Dann ist $\binom{n}{k} \leq \left(\frac{en}{k}\right)^k$

Proposition 2.5.6

$\forall m \geq 1: \frac{2^{2m}}{2\sqrt{m}} \leq \binom{2m}{m} < \frac{2^{2m}}{\sqrt{2m}}$

Prinzip von Inklusion und Exklusion

Satz 2.6.2: Prinzip von Inklusion und Exklusion, Siebformel

Seien $A_1, \ldots, A_n$ endliche Teilmengen eines gemeinsamen Universums. Dann ist

$$\left|\bigcup_{i=1}^n A_i\right| = \sum_{k=1}^n \left((-1)^{k-1} \sum_{I \in \binom{\{1,2,\ldots,n\}}{k}} \left|\bigcap_{i \in I} A_i\right|\right).$$

Notation

$\binom{X}{k}$ bezeichnet die Menge aller k-elementigen Teilmengen von X.

Satz 2.6.4

Die Anzahl der fixpunktfreien Permutationen einer n-elementigen Menge ist $D(n) = \sum_{i=0}^n (-1)^i \frac{n!}{i!}$.

Definition: Eulerfunktion

Die Eulerfunktion φ gibt die Anzahl der zu $n \in \mathbb{N} \setminus \{0\}$ teilerfremden positiven, kleineren natürlichen Zahlen.

$\varphi: \mathbb{N} \setminus \{0\} \to \mathbb{N},\ \varphi(n) = |\{m \in \{1,2,\ldots,n\} \mid ggT(n,m) = 1\}|$

Satz 2.6.5

Sei $n \in \mathbb{N} \setminus \{0\}$ mit der Primfaktorzerlegung $n = p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_r^{\alpha_r}$. Dann ist $\varphi(n) = n\left(1 - \frac{1}{p_1}\right)\left(1 - \frac{1}{p_2}\right) \cdots \left(1 - \frac{1}{p_r}\right)$.

Diskrete Wahrscheinlichkeitsrechnung

Wahrscheinlichkeitsraum

Definition: (Elementar-) ereignisse und Elementarereignismenge Ω

Wir nennen jedes mögliche Ergebnis eines Zufallsexperiments ein Ereignis. Die Vereinigung aller Ereignisse ist die Menge der Elementarereignisse Ω. Ein Ereignis kann sich aus mehreren Elementarereignissen zusammensetzen.

Definition: Ereignismenge Σ

Die Ereignismenge Σ ist eine Teilmenge der Potenzmenge von Ω, also $\Sigma \subseteq 2^\Omega$.

Definition 2.7.2: Wahrscheinlichkeitsmaß

Um von Wahrscheinlichkeiten zu sprechen, ordnen wir den Ereignissen A Zahlen $0 \leq p(A) \leq 1$ zu, wobei 0 für unmögliche und 1 für sichere Ereignisse steht.

Sei Ω eine endliche Menge. Eine Abbildung $p: 2^\Omega \to \mathbb{R}$ heißt Wahrscheinlichkeitsmaß, wenn die Kolmogorov-Axiome erfüllt sind.

K1: $\forall A \subseteq \Omega: p(A) \geq 0$.

K2: $p(\Omega) = 1$.

K3: Für inkompatible Ereignismengen A_1, A_2 ($:= A_1, A_2 \subseteq \Omega$ und $A_1 \cap A_2 = \emptyset$) gilt $p(A_1 \dot\cup A_2) = p(A_1) + p(A_2)$

Das Tupel (Ω, p) nennen wir einen Wahrscheinlichkeitsraum.

Proposition 2.7.4

Sei (Ω, p) ein Wahrscheinlichkeitsraum und seien $A, B, A_i \subseteq \Omega$ mit $i = 1, \dots, k$. Dann gilt:

a.) $A \subseteq B \Rightarrow p(A) \leq p(B)$

b.) Ist $A = A_1 \dot\cup A_2 \dot\cup \cdots \dot\cup A_k$ eine Partition von A, so ist $p(A) = \sum_{i=1}^{k} p(A_i)$.

c.) $p(A \cup B) = p(A) + p(B) - p(A \cap B)$

d.) $p(\Omega \setminus A) = 1 - p(A)$

e.) $p\left(\bigcup_{i=1}^{k} A_i\right) \leq \sum_{i=1}^{k} p(A_i)$

Definition: Laplace-Experimente

Wir nennen Zufallsexperimente mit gleichwahrscheinlichen Elementarereignissen Laplace-Experimente, sprechen von einem uniformen Wahrscheinlichkeitsraum und nennen p die Gleichverteilung auf Ω. Dort gilt dann $p(A) = \frac{|A|}{|\Omega|}$.

Bedingte Wahrscheinlichkeiten

Definition: bedingte Wahrscheinlichkeit

Seien A, B Ereignisse mit $p(B) > 0$. Die Wahrscheinlichkeit von A unter der Bedingung B ist definiert als $p(A|B) = \frac{p(A \cap B)}{p(B)}$.

Definition: unabhängig

Zwei Ereignisse A, B heißen unabhängig, wenn $p(A \cap B) = p(A)p(B)$.

Satz 2.7.8: Satz von Bayes

Sind $A, B \subseteq \Sigma$ Ereignisse mit $p(A)p(B) > 0$, so gilt

$p(B)p(A|B) = p(A)p(B|A)$.

Zufallsvariablen

Definition 2.7.11: Zufallsvariable

Sei (Ω, p) ein Wahrscheinlichkeitsraum. Wir nennen eine Abbildung $X: \Omega \to \mathbb{R}$ eine Zufallsvariable X.

Die Zufallsvariable X induziert ein Wahrscheinlichkeitsmaß

$$q: 2^{X(\Omega)} \to \mathbb{R} \quad \text{mit} \quad q(y) = \sum_{\substack{\omega \in \Omega \\ X(\omega) = y}} p(\{\omega\}) \quad \text{für } y \in X(\Omega).$$

Definition: Erwartungswert

Wir bezeichnen den Wert, den die Zufallsvariable im Mittel annimmt, als Erwartungswert $E(X)$ von X. Es gilt

$$E(X) := \sum_{\omega \in \Omega} p(\{\omega\}) X(\omega).$$

Rechenregeln:

$E(X+Y) = E(X) + E(Y)$

$E(\alpha X) = \alpha E(X)$ für $\alpha \in \mathbb{R}$

Definition: Varianz

Die Varianz $V(X)$ misst die Streuung einer Zufallsvariable, also die erwartete quadratische Abweichung einer Messung vom Erwartungswert. Es gilt

$$V(X) := E\left(\underline{(X - E(X))^2}\right).$$

$E(X')$ mit $X': \Omega \to \mathbb{R}$ mit $x \mapsto (X(x) - E(X))^2$

$$E(X') = \sum_{\omega \in \Omega} p(\{\omega\}) (X(\omega) - E(X))^2$$

Proposition 2.7.12

$V(X) = E(X^2) - (E(X))^2$

Proposition: Siebformel von Poincaré-Sylvester

Seien $A_1, \dots, A_m$ Ereignisse einer diskreten Ereignismenge Ω. Dann gilt:

$$p\left(\bigcup_{k=1}^{m} A_k\right) = \sum_{k=1}^{m} \left((-1)^{k+1} \sum_{I \in \binom{\{1,2,\dots,m\}}{k}} p\left(\bigcap_{i \in I} A_i\right) \right).$$

„Wahrscheinlichkeiten aller Schnitte mit ungerade vielen Ereignissen addieren und Wahrscheinlichkeiten aller Schnitte mit gerade vielen Ereignissen subtrahieren.“

Merkregel:

Vereinigung von Ereignissen → Wahrscheinlichkeiten addieren

Schnitt von Ereignissen → Wahrscheinlichkeiten multiplizieren

Graphen

Relationen

Definition 3.1.1: n-stellige Relation

Seien $M_1, \dots, M_n$ Mengen. Wir nennen eine beliebige Teilmenge $R \subseteq M_1 \times \dots \times M_n$ eine n-stellige Relation.

Notationen:

$(x,y) \in R$: xRy „x steht in Relation zu y"

$(x_1,\dots,x_n) \in R$: „$(x_1,\dots,x_n)$ gehört zur Relation R" oder „$(x_1,\dots,x_n)$ hat das Prädikat R"

„Definition": linke/rechte Partner

Sei $R \subseteq M \times N$ eine binäre Relation. Dann bezeichnen wir zu jedem x, y in R folgende Mengen

$[x]_l := \{ y \in N \mid (x,y) \in R \}$ Menge aller rechten Partner von $x \in M$

$[y]_r := \{ x \in M \mid (x,y) \in R \}$ Menge aller linken Partner von $y \in N$

Bei eindeutiger Interpretation kann der Index weggelassen werden, z.B. bei $[x]_l = [x]_r$.

Definition 3.1.3 Eigenschaften binärer Relationen

Sei $R \subseteq M \times M$ eine binäre Relation. Die Relation heißt

- <u>reflexiv</u>, wenn $\forall x \in M: (x,x) \in R$
- <u>symmetrisch</u>, wenn $(x,y) \in R \Rightarrow (y,x) \in R$
- <u>transitiv</u>, wenn $((x,y) \in R \wedge (y,z) \in R) \Rightarrow (x,z) \in R$
- <u>antisymmetrisch</u>, wenn $((x,y) \in R \wedge (y,x) \in R) \Rightarrow x = y$
- <u>irreflexiv</u>, wenn $\forall x \in M: (x,x) \notin R$

! Irreflexiv und antisymmetrisch sind <u>keine</u> Negationen von reflexiv und symmetrisch!

„Definition": Äquivalenzrelation

Wir nennen eine reflexive, symmetrische und transitive Relation $R \subseteq M \times M$ eine Äquivalenzrelation.

„Definition": Äquivalenzklassen

Äquivalenzrelationen zerlegen die Grundmenge in paarweise disjunkte Äquivalenzklassen, die Mengen $[x]$.

Proposition 3.1.7

Sei R eine Äquivalenzrelation auf M. Dann gilt

a.) $[x] \neq \emptyset$ für alle $x \in M$

b.) Für je zwei $x, y \in M$ ist (entweder) $[x] = [y]$ oder $[x] \cap [y] = \emptyset$. Somit bilden die Äquivalenzklassen eine Partition von M.

c.) R ist durch ihre Äquivalenzklassen vollständig bestimmt.

Definition 3.1.9: Partialordnung

Sei M eine Menge. Wir nennen eine reflexive, antisymmetrische und transitive Relation $R \subseteq M \times M$ eine (ggf. endliche) Partialordnung.

Notationen:

R Partialordnung und $(x,y) \in R$: $x \leq y$

Partialordnung mit Grundmenge P : $(P, \leq)$

„Definition": lineare Ordnung / Totalordnung / Ordnung

Stehen je zwei Elemente in Relation, also für $x, y \in M$ stets $x \leq y$ oder $y \leq x$, so sprechen wir von einer linearen Ordnung, einer Totalordnung oder einfach von einer Ordnung, wenn die Relation bereits eine Partialordnung ist.

Notation: <

Sei $(P, \leq)$ eine Partialordnung mit $a, b \in P$ und $a \leq b$.
Ist $a \neq b$, so schreiben wir $a < b$.

Notation: Bedecken, $<\cdot$

Sei $(P, \leq)$ eine Partialordnung und seien $a, b \in P$ mit $a \leq b$.
Wir sagen b bedeckt a und schreiben $a <\cdot\, b$, wenn gilt:

$$a < b \text{ und } \forall c \in P: a \leq c \leq b \Rightarrow c \in \{a, b\}$$

„Zwischen a und b liegen keine anderen Elemente."

Proposition 3.1.13: Erzeugung endlicher Partialordnungen durch Bedeckungen.

Sei $(P, \leq)$ eine endliche Partialordnung und $x, y \in P$.
Dann gilt

$$x < y \Leftrightarrow \exists x_1, \dots, x_k \in P : x <\cdot\, x_1 <\cdot \dots <\cdot\, x_k <\cdot\, y.$$

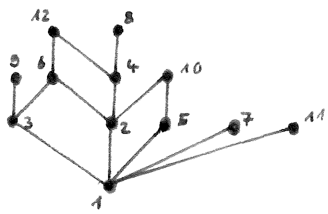
! Zur Beschreibung einer endlichen Partialordnung genügt also die Betrachtung der Bedeckungsrelation!

Visualisierung: Hasse-Diagramm

Endliche Partialordnungen können als Hasse-Diagramme dargestellt werden. Dabei werden Elemente als Punkte und die Relationen als Verbindungen dargestellt, wobei Verbindungen zwischen transitiven Punkten nicht eingezeichnet werden. Die Punkte werden von unten nach aufsteigendem Wert angeordnet.

Beispiel:

Teilbarkeitsrelation $a \leq b \Leftrightarrow a$ teilt b auf der Menge $\{1, 2, \dots, 12\}$



Graphen

Definition 3.2.1 Graphen

Sei V eine endliche Menge („vertices"/Knoten) und $E \subseteq \binom{V}{2}$ Teilmenge der zweielementigen Teilmengen von V („edges"/Kanten).
Wir nennen das geordnete Paar (V, E) einen (einfachen, ungerichteten) Graphen.

Notation:

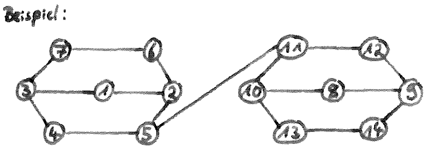
Sei G ein Graph.

$V(G) :=$ Knotenmenge von G

$E(G) :=$ Kantenmenge von G

$\{u, v\} \in E(G) \Rightarrow$ „u und v sind adjazent" oder „u und v sind Nachbarn" oder „u kennt v und umgekehrt"

Beispiel:



$V = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14\}$

$E = \{\{1,2\}, \{1,3\}, \{2,6\}, \{6,7\}, \{3,7\}, \{3,4\}, \{4,5\}, \{2,5\}, \{5,11\}, \{11,12\}, \{9,12\}, \{9,14\}, \{13,14\}, \{10,13\}, \{10,11\}, \{8,9\}, \{8,10\}\}$

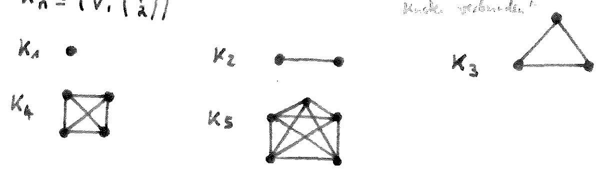
„Definition": Graphenklassen

Sei $V = \{1, 2, \dots, n\}$ mit $n \in \mathbb{N}$.

- vollständiger Graph K_n mit n Knoten

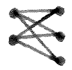
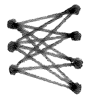
  $K_n = (V, \binom{V}{2})$

  „Jeder Knoten ist mit allen anderen Knoten verbunden."



- Kreis C_n mit n Knoten
  Sei $n \geq 3$. C_n hat Knotenmenge V und Kantenmenge $\{i, i+1\}$ für $i = 1, \ldots, n-1$ und $\{n, 1\}$.
  C_3  C_4  C_5 

- Weg P_n mit n Knoten
  Knotenmenge V, Kantenmenge $\{i, i+1\}$ für $i = 1, \ldots, n-1$.
  P_1 P_2 P_3 

- Vollständiger, bipartiter Graph $K_{m,n}$ mit $m+n$ Knoten
  Knotenmenge $V \cup W$ mit $W = \{n+1, \ldots, n+m\}$,
  Kantenmenge $V \times W$
  „Jeder Knoten aus V ist mit jedem Knoten aus W verbunden"
  $K_{2,3}$  $K_{3,4}$ 

__Definition 3.2.4: Isomorphismus__

Seien $G = (V, E)$ und $G' = (V', E')$ zwei Graphen. Wir nennen G und G' isomorph und schreiben $G \cong G'$, wenn* es eine bijektive Abbildung $f: V \to V'$ gibt mit
$$\forall u, v \in V: \{u, v\} \in E \iff \{f(u), f(v)\} \in E'.$$
Die Abbildung f heißt dann Isomorphismus.

__„Proposition:" Anzahl nicht isomorpher Knoten__

Es gibt mindestens $\frac{2^{\binom{n}{2}}}{n!}$ paarweise nicht isomorphe Graphen mit n Knoten. Die Anzahl paarweise nicht isomorpher Graphen ist laut S.69
$$\Omega(2^{\frac{n^2}{2}(1-\varepsilon)}) \text{ für jedes } \varepsilon > 0.$$

__Definition 3.2.9: Multigraph__

Ein Multigraph $G = (V, E, ad)$ ist ein Tripel bestehend aus einer endlichen Knotenmenge V, einer endlichen Kantenmenge E und einer Adjanzenfunktion $ad: E \to \binom{V}{2} \cup V$, die jeder Kante einen oder zwei Endknoten zuordnet.

Haben $e, e' \in E$ die gleichen Endknoten, so heißen sie parallel. Eine Kante mit nur einem Endknoten heißt Schleife.

__Definition 3.3.1: (induzierter) Teilgraph__

Seien $G = (V, E)$ und $H = (W, F)$ Graphen. Wir nennen H einen Teilgraph von G, wenn $W \subseteq V$ und $F \subseteq E$.

Wir nennen H einen induzierten Teilgraph, wenn darüber hinaus $F = E \cap \binom{W}{2}$ gilt.

„induzierter Teilgraph = Teilmenge der Knoten + __alle__ Kanten zwischen diesen Knoten."

__„Definition": Weg / Pfad__ „Keine Wiederholungen von Knoten oder Kanten"

Wir nennen ein Teilgraph, der isomorph zu einem Weg P_ℓ ist, Weg oder Pfad im Graphen. Ein Weg kann unterschiedlich dargestellt werden

- alternierende Knoten- und Kantensequenz (__paarweise verschieden__) !
  $(v_0, e_1, v_1, e_2, \ldots, e_k, v_k)$ mit $e_i = (v_{i-1}, v_i)$
- Knotensequenz (paarweise verschieden)
  $(v_0, v_1, \ldots, v_k)$
- Kantensequenz (paarweise verschieden)
  $(e_1, e_2, \ldots, e_k)$

v_0-v_k-Weg der Länge k

__„Definition": Kreis__

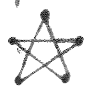
Wir nennen einen Teilgraphen, der isomorph zu einem Kreis ist, einen Kreis im Graphen.

Darstellung analog Weg (s.o.).

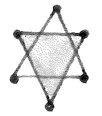
__Definition 3.4.1: zusammenhängender Graph, (Zusammenhangs-) Komponente__

Ein Graph $G = (V, E)$ heißt zusammenhängend, wenn es zu je zwei Knoten u, v einen u-v-Weg gibt. Ein mengentheoretisch maximaler zusammenhängender Teilgraph eines Graphen heißt Komponente oder Zusammenhangskomponente.

Beispiel:



Pentagramm: zusammenhängend



Davidstern:
- nicht zusammenhängend
- hat zwei Komponenten

Nachteil Definition:
Wir müssen bei der Definition über Weg darauf achten, dass die Wege keine Wiederholung von Knoten oder Kanten haben.

__Definition 3.4.3: Spaziergang__

Sei $k \in \mathbb{N}$. Wir nennen eine alternierende Folge von Knoten und Kanten $(v_0, e_1, v_1, e_2, \ldots, e_k, v_k)$ mit $e_i = (v_{i-1}, v_i)$ einen Spaziergang der Länge k von v_0 nach $v_\ell = v_k$.

__Proposition 3.4.4__

Sei $G = (V, E)$ ein Graph und $v_0, v_\ell \in V$. Dann gilt:
Es gibt genau dann einen v_0-v_ℓ-Weg, wenn es einen Spaziergang von v_0 nach v_ℓ gibt.

Anm.: Aufgrund diese Tatsache definieren wir die Knoten der Komponenten als Äquivalenzklassen der Äquivalenzrelation (!)
$a R b \iff$ es gibt einen Spaziergang von a nach b

__Definition 3.4.6: Abstand__

Sei $G = (V, E)$ ein zusammenhängender Graph und $u, v \in V$. Die Länge eines kürzesten u-v-Weges nennen wir den Abstand $dist_G(u, v)$ von u und v in G.
Abstandsfunktion / Metrik $dist_G(u, v): V \times V \to \mathbb{N}$.

__Proposition 3.4.7: Eigenschaften einer Metrik__

Die Metrik eines Graphen erfüllt
- Nichtnegativität und Definitheit
  $dist_G(u, v) \geq 0$ und $(dist_G(u, v) = 0 \iff u = v)$
- Symmetrie
  $\forall u, v \in V: dist_G(u, v) = dist_G(v, u)$
- Dreiecksungleichung
  $\forall u, v, w \in V: dist_G(u, w) \leq dist_G(u, v) + dist_G(v, w)$.

__Kodierung von Graphen__

__Definition 3.5.1: Adjazenz- und Inzidenzmatrix__

Sei $G = (V, E)$ ein Graph mit Knotenmenge $V = \{v_1, \ldots, v_n\}$ und Kantenmenge $E = \{e_1, \ldots, e_m\}$. Die

Die Adjazenzmatrix $A_G = (a_{ij})$ ist eine $n \times n$-Matrix definiert vermöge
$$a_{ij} = \begin{cases} 1, & \text{wenn } (v_i, v_j) \in E \\ 0 & \text{sonst} \end{cases}$$

Die Knoten-Kanten Inzidenzmatrix $B_G = (b_{ij})$ ist eine $n \times m$-Matrix definiert vermöge
$$b_{ij} = \begin{cases} 1, & \text{wenn } v_i \text{ Endknoten von } e_j \text{ ist} \\ 0 & \text{sonst} \end{cases}$$

! Matrizendarstellung eher von theoretischem Interesse, da fast ausschließlich Nullen gespeichert werden!

__Proposition 3.5.3 Anzahl der Spaziergänge von v_i nach v_j der Länge k__

Sei $G = (V, E)$ ein Graph mit Knotenmenge $V = \{v_1, \ldots, v_n\}$ und sei $A = A_G$ seine Adjazenzmatrix. Für $k \in \mathbb{N}$ bezeichnen wir mit A^k die k-te Potenz der Adjazenzmatrix bzgl. der Matrizenmultiplikation. Bezeichne a_{ij}^k den Eintrag der Matrix A^k an der Stelle (i, j). Dann ist a_{ij}^k die Anzahl der Spaziergänge von v_i nach v_j der Länge k.

Korollar 3.5.4

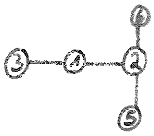
Der Abstand zweier Knoten v_i, v_j ist $dist_G(v_i, v_j) = \min\{k \in \mathbb{N} \mid a_{ij}^k \neq 0\}$.

„Definition": Adjazenzliste

Für jeden Knoten des Graphen werden die mit ihm durch Kanten direkt verbundenen Knoten gespeichert (z.B. in einer doppelt verketteten Liste).

Beispiel

Knoten	1	2	…
Adjazenzliste	3 2	1 5 6	



Ausführungen zu effizienten Algorithmen

Siehe S. 77 - 79.

Breitensuche (BFS / Breadth-First-Search)

Sei $G = (V, E)$ ein Graph und $v \in V$ ein Startknoten. Wir haben die Komponente von v zu bestimmen („Welche Knoten sind von v aus zu erreichen?").
Dazu ermitteln wir iterativ die Nachbarknoten des jeweils aktuellen Knotens.

Datenstrukturen:
- pred[x] für alle $x \in V$; speichert Vorgängerknoten von Knoten x
- component[x] für alle $x \in V$; speichert Komponente des Knotens x
- Q Liste für noch zu betrachtende Knoten des Graphen

Beschreibung Algorithmus:
- Phase 1: Initialisierung
  Setze pred[v] = v und component[v] = v für Startknoten v
  Lege v auf Stack Q
- Phase 2: Iteration
  Solange Stack nicht leer:
  Erstes Stackelement nehmen
  Nachbarknoten testen:
  Wenn Nachbar noch keinen Vorgänger
  Vorgänger = aktueller Knoten
  Komponente Nachbar = Komponente aktueller Knoten
  Nachbar auf Stack

Algorithmus:

```
for v in Vertices:        } Erweiterung um alle Komponenten unabhängig von
  if pred[v] == None:     } einem Startpunkt v∈V zu bestimmen
    pred[v] = v
    component[v] = v
    Q.Append(v)
  while Q.IsNotEmpty():
    v = Q.Top()
    for w in Neighborhood(v):
      if pred[w] == None:
        pred[w] = v
        component[w] = component[v]
        Q.Append(w)
```

Die einzelnen Komponenten nennen wir Breitensuchbaum (BFS-tree).

Satz 3.7.1

BFS berechnet die Komponenten eines Graphen in $O(|V| + |E|)$ Zeit.

Tiefensuche (DFS, Depth-First-Search)

Sei $G = (V, E)$ ein Graph und $v \in V$ ein Startknoten. Wir haben die Komponente von v zu bestimmen und die Knotennummerierung (Entdeckungsreihenfolge) zu berechnen.
Dazu suchen wir die Nachbarn von v und setzen die Suche beim ersten gefundenen Nachbarn fort.

Datenstrukturen:
- pred[x] für alle $x \in V$; speichert Vorgängerknoten von Knoten x
- label[x] für alle $x \in V$; speichert Knotennummerierung
- step: globale Variable für aktuell freie Knotennummer

Algorithmus (rekursiv):

```
def DFS(v):
  global step
  label[v] = step
  step = step + 1
  for w in Neighborhood(v):
    if label[w] == None:
      pred[w] = v
      DFS(w)
  return
step = 1
DFS(r)
```

Satz 3.8.1

Die Prozedur DFS terminiert in $O(|V| + |E|)$ Zeit.

Valenzsequenzen

„Definition": Knotengrad / Valenz

Sei $G = (V, E)$ ein Graph oder Multigraph mit $v \in V$. Wir nennen die Anzahl der Kanten, deren Endknoten v ist, den Knotengrad oder die Valenz $deg_G(v)$ von v. In Multigraphen werden Schleifen doppelt gezählt.

„Definition": Gradsequenz / Valenzsequenz

Ist $v_1, \dots, v_n$ eine lineare Anordnung der Knoten von V, so nennen wir $(deg(v_1), \dots, deg(v_n))$ die Gradsequenz oder Valenzsequenz des Graphen.
Können zwei Valenzsequenzen durch Umordnen der Valenzen ineinander überführt werden, so sehen wir sie als gleich an. Wir nehmen eine absteigende Sortierung an.

Proposition 3.9.1 Handshake-Lemma

In jedem Graphen oder Multigraphen $G = (V, E)$ ist die Summe der Knotengrade gerade, genauer $\sum_{v \in V} deg(v) = 2|E|$.

Korollar 3.9.2

In jedem Graphen oder Multigraphen ist die Anzahl der Knoten mit ungeradem Knotengrad gerade.

Satz 3.9.3 Kriterium für Charakterisierung eines Graphen durch Gradfolgen

Sei $d_1 \geq d_2 \geq \dots \geq d_n \geq 0$ eine Folge natürlicher Zahlen. Dann gilt:
$(d_1, \dots, d_n)$ ist Gradsequenz eines einfachen Graphen
$\iff \sum_{i=1}^n d_i$ gerade und $\forall i \in \{1, \dots, n\}: \sum_{j=1}^i d_j \leq i(i-1) + \sum_{j=i+1}^n \min\{i, d_j\}$.

Aufgabe 3.9.5 Kriterium für Charakterisierung eines Multigraphen durch Gradfolgen

Sei $d_1 \geq d_2 \geq \dots \geq d_n \geq 0$ eine Folge natürlicher Zahlen. Dann gilt
$(d_1, \dots, d_n)$ ist Gradsequenz eines Multigraphen
$\iff \sum_{i=1}^n d_i$ gerade.

Satz 3.9.6

Sei $D = (d_1, d_2, \dots, d_n)$ eine Folge natürlicher Zahlen mit $n > 1$ und $d_1 \geq d_2 \geq \dots \geq d_n \geq 0$. Dann gilt:

D ist Valenzsequenz eines einfachen Graphen $\iff$

$d_1 + 1 \leq n$ und

Folge $D' = (d'_2, d'_3, \dots, d'_n)$ mit $d'_i := \begin{cases} d_i - 1 & \text{für } i = 2, \dots, d_1 + 1 \\ d_i & \text{für } i = d_1 + 2, \dots, n \end{cases}$

ist die Valenzsequenz eines einfachen Graphen.

Verfahren von Havel und Hakimi

Aus Satz 3.9.6 können wir unmittelbar folgendes Verfahren ableiten, um zu entscheiden, ob eine Sequenz die Valenzsequenz eines Graphen ist:

Iteriere

- Sequenz nach absteigenden Knotengraden anordnen
- Erstes Element x aus der Sequenz entfernen und alle x folgenden Elemente um 1 dekrementieren

Eulertouren

Definition 3.10.1: Eulertour

Sei $G = (V, E)$ ein Multigraph ohne isolierte Knoten, also $G = K_1$ oder $\deg(v) > 0$ für alle $v \in V$. Sei $v_0 \in V$. Wir nennen einen Spaziergang $v_0 e_1 v_1 e_2 \cdots e_m v_0$ von v_0 nach v_0 Eulertour, wenn er jede Kante* genau einmal benutzt. *des Graphen

Der Graph G heißt eulersch, wenn er eine Eulertour ausgehend von einem (und damit von jedem) Knoten $v_0 \in V$ hat.

Satz 3.10.2

Sei $G = (V, E)$ ein Multigraph. Dann sind paarweise äquivalent:

a.) G ist eulersch

b.) G ist zusammenhängend und alle Knoten haben geraden Knotengrad

c.) G ist zusammenhängend und E ist eine disjunkte Vereinigung von Kreisen

Algorithmus Eulertour

Folgender Algorithmus kann aus Satz 3.10.2 abgeleitet werden

Idee: Suche Kreis („Tour") in G. Suche bei allen Knoten der ermittelten Tour mit ungenutzten Kanten nach weiteren Touren und füge sie mit dem Startknoten der Ausgangstour hinzu.

Notation:

- T[]: speichert Eulertour als Kantenliste
- HatT[]: speichert aktuelle Tour als Kantenliste
- T.CurrentVertex(): aktueller Knoten in T mit unbenutzten Kanten
- T.SpliceIntoCurrentPosition(HatT): fügt HatT in T an der Stelle T.CurrentVertex() ein
- T.IncPosition(): inkrementiert T.CurrentVertex()
- T.PositionAtEnd(): Ende von T erreicht?
- NextEdge(vertex): liefert nächste Kante von vertex und löscht sie aus den Adjazenzlisten beider Endknoten.

Algorithmus:

```
Vertex = PickVertex()
done = False
e = NextEdge(vertex)
while not done:
    while e:
        HatT.Append(e)
        vertex = OtherEnd(e, vertex)
        e = NextEdge(vertex)
    T.SpliceIntoCurrentPosition(HatT)
    vertex = T.CurrentVertex()
    while True:
        e = NextEdge(vertex)
        if not e:
            T.IncPosition();
            if T.PositionAtEnd:
                done = True
                break
            vertex = T.CurrentVertex()
        else:
            break
```

nstruktion von
mer ausgehd.
an einem Startpunkt

ellange die bisherige Tour
en die Knoten und suche
ne ungenutzte
ante

Satz 3.10.4

In einem eulerschen Graph $G = (V, E)$ kann man eine Eulertour in $O(|V| + |E|)$ bestimmen.

Gerichtete Graphen und Eulertouren

„Definition": Diagonale Δ

Sei V eine Menge. Wir bezeichnen die Menge der Diagonalelemente von $V \times V$ mit

$$\Delta := \{(v, v) \mid v \in V\}.$$

Definition 3.11.1: gerichtete Graphen / Digraphen

Sei V eine endliche Knotenmenge und sei $A \subseteq (V \times V) \setminus \Delta$.

Wir nennen das geordnete Paar (V, A) einen einfachen, gerichteten Graphen oder einen Digraphen.

Wir nennen die Kante $(v, w) \in A$ Bogen, v den Anfang (tail) und w das Ende (head).

„Definition": Multidigraphen

Wir nennen einen Digraphen einen Multidigraphen, wenn gleichgerichtete Kanten mehrmals vorkommen dürfen oder Schleifen vorkommen.

„Definition": gerichteter Spaziergang

Ein gerichteter Spaziergang ist eine alternierende Folge aus Knoten und Bögen $(v_0, a_1, v_1, a_2, \dots, a_k, v_k)$ mit $a_i = (v_{i-1}, v_i)$.

„Definition": Innengrad und Außengrad

Sei $G = (V, E)$ ein Digraph. Dann bezeichnet

a.) der Innengrad $\deg^+_G(v)$ die Anzahl der einlaufenden Kanten mit Ende $v \in V$ und

b.) der Außengrad $\deg^-_G(v)$ die Anzahl der auslaufenden Kanten mit Anfang $v \in V$.

„Definition": zugrundeliegender Multigraph

Sei G ein (Multi-)Digraph. Der Multigraph, der durch Weglassen der Orientierung der Bögen entsteht, wird zugrundeliegender Multigraph genannt.

„Definition": Orientierung des zugrundeliegenden Graphen

Ist der zugrundeliegende Multigraph ein Graph, so heißt der Digraph eine Orientierung des zugrundeliegenden Graphen.

„Definition": zusammenhängender Digraph

Ein Digraph heißt zusammenhängend, wenn der zugrundeliegende ungerichtete Graph zusammenhängend ist.

Definition 3.11.2: Eulertour

Sei $D = (V, A)$ ein (Multi-)Digraph ohne isolierte Knoten. Ein Spaziergang, der jeden Bogen genau einmal benutzt und in seinem Anfangsknoten endet, heißt Eulertour.

Ein (Multi-)Digraph heißt eulersch, wenn er eine Eulertour hat.

Satz 3.11.3

Sei $D = (V, A)$ ein (Multi-)Digraph. Dann sind paarweise äquivalent:

a.) D ist eulersch

b.) D ist zusammenhängend und alle Knoten haben gleichen Innen- wie Außengrad

c.) D ist zusammenhängend und A ist disjunkte Vereinigung von gerichteten Kreisen

2-Zusammenhang

Definition 3.12.1: k-zusammenhängend

Sei $G=(V,E)$ ein Graph und sei $k \geq 2$.
Wir nennen G k-fach knotenzusammenhängend oder kurz k-zusammenhängend, wenn $|V| \geq k+1$ ist und der Graph nach Entfernen beliebiger $k-1$ Knoten immer noch zusammenhängend ist.
Wir nennen G k-fach kantenzusammenhängend, wenn der Graph nach Entfernen beliebiger $k-1$ Kanten immer noch zusammenhängend ist.

Definition 3.12.1: Knoten-/Kantenzusammenhangszahl

Sei $G=(V,E)$ ein Graph. Wir nennen die größte natürliche Zahl, für die G knotenzusammenhängend ist, die Knotenzusammenhangszahl $\kappa(G)$.
Wir nennen die größte natürliche Zahl, für die G kantenzusammenhängend ist, die Kantenzusammenhangszahl $\kappa'(G)$.

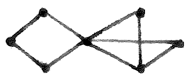
Definition 3.12.2: Operationen auf Graphen

Sei $G=(V,E)$ der nachfolgende Graph.

Wir definieren folgende (Multi-)Graphen, die durch Operationen auf G entstehen:

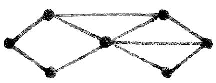
Entfernen einer Kante $e \in E$

$G \setminus e := (V, E \setminus \{e\})$



Einfügen einer Kante $\bar{e} \in \binom{V}{2} \setminus E$

$G + \bar{e} := (V, E \cup \{\bar{e}\})$

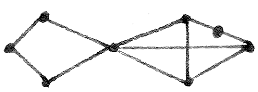


Entfernen eines Knotens $v \in V$

$G \setminus v := (V \setminus \{v\}, \tilde{E})$ mit $\tilde{E} := \{e \in E \mid v \notin e\}$

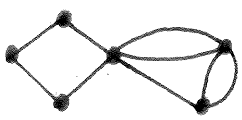
Unterteilen einer Kante $e \in E$

Sei $e=(v,w)$.
$G \% e := (V \cup u, \hat{E})$ mit $u \notin V$ und
$\hat{E} := (E \setminus \{e\}) \cup \{(v,u),(u,w)\}$



Kontraktion einer Kante $e \in E$

Sei $e=(v,w)$. Die Kontraktion von e ist der Multigraph
$G/e = (\tilde{V}, \tilde{E})$ mit $\tilde{V} := V \cup \{u\} \setminus \{v,w\}$, wobei $u \notin V$ ein neuer Knoten ist, und
$\tilde{E} := \{e \in E \mid e \cap \{v,w\} = \emptyset\} \cup \{(u,x) \mid (v,x) \in E\}$
$\cup \{(y,u) \mid (y,w) \in E\}$.



Anschauliche Beschreibung: Kante e ist ein elastisches Band, das solange zusammengezogen wird, bis beide Endknoten zusammenliegen. Anschließend wird e entfernt.

Satz 3.12.3

Ein Graph ist 2-knotenzusammenhängend $\iff$
je zwei Knoten $u \neq v$ liegen auf einem gemeinsamen Kreis

Verallgemeinerung von Satz 3.12.3: Satz von Menger

Ein Graph ist k-knotenzusammenhängend $\iff$
zu je zwei Knoten u, v gibt es k u-v-Wege, die paarweise nur die Endknoten gemeinsam haben.

Korollar 3.12.5

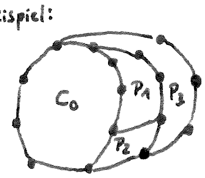
Ein Graph $G=(V,E)$ ist 2-zusammenhängend $\iff$
jede Unterteilung von G ist 2-zusammenhängend.

Definition 3.12.6: Ohrenzerlegung von G

Sei $G=(V,E)$ ein Graph. Eine Folge $(C_0, P_1, P_2, \dots, P_k)$ heißt Ohrenzerlegung von G, wenn gilt:

a.) C_0 ist ein Kreis
b.) $\forall i = 1, \dots, k$: P_i ist ein Pfad, der mit $V(C_0) \cup \bigcup_{j=1}^{i-1} V(P_j)$ genau seinen Anfangs- und Endknoten gemeinsam hat.
c.) $E(C_0), E(P_1), \dots, E(P_k)$ bilden eine Partition der Kantenmenge E.

Beispiel:



Satz 3.12.8

Ein Graph G ist 2-zusammenhängend $\iff$ G hat eine Ohrenzerlegung

Bäume und Matchings

Definition und Charakterisierung

Definition 4.1.1: Baum

Wir nennen einen zusammenhängenden Graph $T=(V,E)$, der keinen Kreis enthält, einen Baum („tree").

Definition 4.1.2: Blatt

Sei $G=(V,E)$ ein Graph und $v \in V$ mit $\deg(v) = 1$. Dann nennen wir v ein Blatt von G.

Lemma 4.1.3

Jeder Baum mit mindestens zwei Knoten hat mindestens zwei Blätter.

Lemma 4.1.4

Sei $G=(V,E)$ ein Graph und v ein Blatt in G. Dann gilt:
G ist ein Baum $\Leftrightarrow$ $G \setminus v$ ist ein Baum.

„Von einem Baum können beliebige Blätter entfernt oder hinzugefügt werden."

Satz 4.1.5

Sei $T=(V,E)$ ein Graph mit $|V| \geq 2$. Dann sind paarweise äquivalent:

a.) T ist ein Baum.
b.) Zwischen je zwei Knoten $v,w \in V$ gibt es genau einen Weg von v nach w. („eindeutige Wege")
c.) T ist zusammenhängend und für alle $e \in E$ ist $T \setminus e$ unzusammenhängend. („1-Zusammenhang")
d.) T ist kreisfrei und für alle $\bar{e} \in \binom{V}{2} \setminus E$ enthält $T+\bar{e}$ einen Kreis.
e.) T ist zusammenhängend und $|E| = |V|-1$.
f.) T ist kreisfrei und $|E| = |V|-1$.

Definition 4.1.7: Fundamentalkreis

Sei $T=(V,E)$ ein Baum. Wir bezeichnen den eindeutigen Kreis C in T, der von $\bar{e} \in \binom{V}{2} \setminus E$ geschlossen wird, als Fundamentalkreis $C(T,\bar{e})$ bezüglich T und $\bar{e}$.

„Proposition": Aufgabe 4.1.6

Sei $T=(V,E)$ ein Baum und $C(T,\bar{e})$ ein Fundamentalkreis. Dann gilt:
$\forall e \in C(T,\bar{e}) \setminus \bar{e}$: $(T+\bar{e}) \setminus e$ ist ein Baum.

„Ein durch Kante $\bar{e}$ erzeugter Fundamentalkreis eines Baums T kann durch Entfernen einer Kante $\neq \bar{e}$ wieder in einen Baum gewandelt werden."

Isomorphismen von Bäumen

Definition 4.2.1: Wurzelbaum / Arboreszenz

Sei $T=(V,E)$ ein Baum und $r \in V$ ein ausgezeichneter Knoten, den wir als Wurzelknoten bezeichnen. Das Paar (T,r) nennen wir Wurzelbaum oder Arboreszenz. Dabei denken wir uns alle Kanten des Baumes so orientiert, dass die Wege von r zu allen anderen Knoten v gerichtete Wege sind. Ist dann (v,w) ein Bogen, so nennen wir v den Elternknoten von w und w Kind oder direkten Nachfahren von v.

Definition 4.2.3: gepflanzter Baum

Ein gepflanzter Baum (T,r,p) ist ein Wurzelbaum, bei dem an jedem Knoten $v \in V$ eine Reihenfolge $p(v)$ der direkten Nachfahren vorgegeben ist. Dadurch ist eine „Zeichenvorschrift" definiert, wie wir den Graphen in die Ebene einzubetten haben.

„Definition": Isomorphismus von Wurzelbäumen

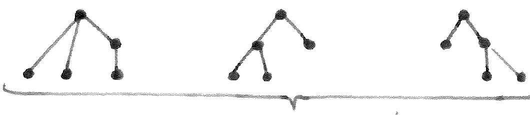
Seien $T=(V,E)$ und $T'=(V',E')$ Bäume mit Wurzelbäumen (T,r) und (T',r'). Wir nennen die Wurzelbäume isomorph, wenn T und T' isomorph sind und zusätzlich r auf r' abgebildet wird.

$(T,r) \cong (T',r')$ $\Leftrightarrow$ Bijektion $f: V \to V'$ mit $f(r)=r'$
$\forall u,v \in V$: $\{u,v\} \in E \Leftrightarrow \{f(u), f(v)\} \in E'$

„Definition": Isomorphismus gepflanzter Bäume

Isomorphismus der Wurzelbäume mit Berücksichtigung der Reihenfolge der direkten Nachfahren.

Beispiel: Isomorphismus



Bäume paarweise isomorph

als Wurzelbäume isomorph

gepflanzte Bäume paarweise nicht isomorph

„Proposition": Aufgabe 4.2.2

Sei $D=(V,A)$ ein zusammenhängender, gerichteter Graph. Dann gilt:
D ist ein Wurzelbaum $\Longleftrightarrow$ $\exists!\, r \in V$: $\deg^{+}(r) = 0$ und
$\forall v \in V \setminus \{r\}$: $\deg^{+}(v) = 1$.

„Definition": lexikographische Ordnung

Sei $(\Sigma, \leq)$ eine endliche, total geordnete Menge („Alphabet"). Wir nennen $w = (w_1, w_2, \dots, w_m) \in \Sigma^m$ mit $m \in \mathbb{N}$ ein Wort über Σ. Sei Σ^* die Menge aller Wörter beliebiger Länge über Σ. Dann ist die lexikographische Ordnung auf Σ^* die Totalordnung

$(\Sigma^*, \preceq)$ mit

$$\forall w, \tilde{w} \in \Sigma^*: w \preceq \tilde{w} \Longleftrightarrow \exists k \in \mathbb{N}\ \forall 1 \leq i \leq k: w_i = \tilde{w}_i \text{ und } \begin{cases} w \in \Sigma^k, \tilde{w} \in \Sigma^m \text{ mit } m \geq k \text{ oder} \\ w_{k+1} < \tilde{w}_{k+1} \end{cases}$$

„(Wörter identisch) oder (ein Wort ist Anfang des anderen Worts) oder (Wörter haben gleichen Anfang und werden danach unterschiedlich fortgesetzt)"

Verfahren zum Nachweis der Isomorphie zweier Bäume

Satz 4.2.11

Zwei Bäume haben genau dann den gleichen Code, wenn sie isomorph sind.

Verfahren

Wir wenden auf beide Bäume folgende Schritte an, um jeweils den Code zu bestimmen und anschließend miteinander zu vergleichen.

a.) Bestimme zum gegebenen Baum eine Wurzel
b.) Bestimme zum gegebenen Wurzelbaum eine kanonische Pflanzung (Anordnungsreihenfolge der Kinder aller Knoten)
c.) Bestimme zum gegebenen gepflanzten Baum den eindeutigen Code („Code des kanonisch gewurzelten und kanonisch gepflanzten Baums")

a.) Isomorphismen von Bäumen

Finde zum gegebenen Baum ein Wurzelknoten, der unter Isomorphismen fix bleibt.

Definition 4.2.9: Exzentrizität / Zentrum

Sei $G=(V,E)$ ein Graph und $v \in V$. Wir bezeichnen die Zahl $ex_G(v) = \max\{dist_G(v,w) \mid w \in V\}$ als Exzentrizität.
Als Zentrum $Z(G)$ bezeichnen wir die Menge der Knoten minimaler Exzentrizität
$Z(G) = \{v \in V \mid ex_G(v) = \min\{ex_G(w) \mid w \in V\}\}$.

Das Zentrum eines Graphen ist eine Isomorphieinvariante.

Lemma 4.2.10

Sei $T=(V,E)$ ein Baum. Es gilt
1.) $|Z(T)| \leq 2$
2.) $Z(T) = \{x,y\}$ mit $x \neq y$ $\Rightarrow$ $(x,y) \in E$.

Verfahren:

- Bestimme das Zentrum des Baums
  - ↳ entferne solange Blätter, bis nur ein oder zwei Knoten übrig bleiben
- Fall 1: $Z(T) = \{x\}$
  - ↳ x ist Wurzel von T
- Fall 2: $Z(T) = \{x_1, x_2\}$
  - ↳ entferne Kante $\{x_1, x_2\}$
  - ↳ bestimme Codes der in x_1 bzw. x_2 gewurzelten Teilbäume
    - ↳ Fall 1: Codes sind gleich
      - ↳ wähle beliebigen Knoten
    - ↳ Fall 2: Codes sind nicht gleich
      - ↳ wähle Knoten, dessen Code des Teilbaums lexikographisch kleiner ist.

      Definiere „(" < „)", also

      $A \preceq B$, wenn $\begin{cases} A \text{ ist Anfang von } B & \text{oder} \\ \text{erste unterscheidbare Klammer in } A \text{ ist offen} \end{cases}$

b.) Isomorphismus von Wurzelbäumen

überführe einen gegebenen Wurzelbaum durch Festlegen der Reihenfolge der Kinder in einen gepflanzten Baum

Verfahren (Bottom-up)

- alle Blätter haben Code ()
- x Knoten mit Kindern mit bekanntem Code $\Rightarrow$ Sortiere Kinder zu $y_1, \dots, y_k$, sodass für die Codes $C_1 \preceq \dots \preceq C_k$ gilt.
- x erhält Code $(C_1 C_2 \dots C_k)$

Aufspannende Bäume

Definitionen 4.3.1

Wir nennen einen kreisfreien Graphen einen Wald.

Teilgraph $T = (V, F)$ ist $G = (V, E)$ **aufspannend** $\Leftrightarrow$
Knotenmenge Zusammenhangskomponenten von T =
Knotenmenge Zusammenhangskomponenten von G

Teilgraph $T = (V, F)$ ist $G = (V, E)$ **aufspannender Wald / Gerüst** $\Leftrightarrow$
Knotenmenge Zusammenhangskomponenten von T =
Knotenmenge Zusammenhangskomponenten von G und
T ist kreisfrei

Teilgraph $T = (V, F)$ ist $G = (V, E)$ **aufspannender Baum** $\Leftrightarrow$
G ist zusammenhängend und T ist ein Baum

Zwei Algorithmen zur Berechnung eines aufspannenden Baum in zusammenhängenden Graphen

Algorithmus 4.3.2

Sei E eine (beliebig sortierte) Kantenliste des Graphen (V, E) und sei zu Anfang $T = \emptyset$.

T.CreatingCycle(e): überprüft zur kreislosen Kantenmenge T mit $e \notin T$, ob $T + e$ einen Kreis enthält.

```
for e in E:
    if not T.CreatingCycle(e):
        T.AddEdge(e)
```

Lemma 4.3.3

Algorithmus 4.3.2 berechnet einen G aufspannenden Wald.

Ausführungen / Analyse

Komplexität des Algorithmus hängt von der effektiven Implementierung des Kreistests ab.

triviale Implementierung
- label von Endknoten u von $e = (u, v)$ alle in T von u aus erreichbaren Knoten
- wird v gelabelt, so schließt e einen Kreis mit T
- $\rightarrow$ Gesamtlaufzeit $O(|V| \cdot |E|)$

Zur Effizienzsteigerung ist nachfolgendes Problem zu lösen:

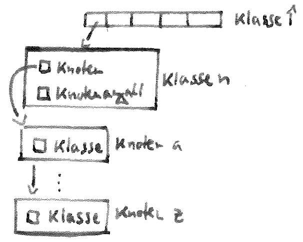
Problem 4.3.4: UNION-FIND

Sei $V = \{1, \dots, n\}$ und eine initiale Partition in n Klassen $V = \{1\} \cup \dots \cup \{n\}$ gegeben. Wir nennen eine Datenstruktur UNION-FIND-Struktur, wenn folgende Operationen ausführbar sind:

UNION: vereinigt Klassen von x, y
FIND: Prüfe, ob $x, y \in V$ in der gleichen Klasse liegen

Anwendung auf Algorithmus 4.3.2
- T besteht zu jedem Zeitpunkt aus den Kanten eines Waldes
- Knotenmengen der Zusammenhangskomponenten liefern Partition
- Kreistest: prüfe, ob Endknoten x, y der Kante (x, y) in der gleichen Klasse liegen

Einfache Lösung



Find: Vergleich der Klassennummern der Knoten
UNION: kleinere Komponente erhält Klassennummer der größeren Komponente, addiere Knotenanzahl der kleineren Komponente auf größere Komponente

Lemma 4.3.5

Die Kosten des Komponentenverschmelzens betragen über den gesamten Lauf des Algorithmus akkumuliert $O(|V| \log |V|)$.

Algorithmus 4.3.7

Allgemeinere Version des BFS
Sei $G = (V, E)$ ein Graph und sei $v \in V$.
- Setze $V_0 = \{v\}$, $T_0 = \emptyset$, $i = 0$
- Solange es geht
  - wähle Kante $e = (x, y) \in E$ mit $x \in V_i$, $y \notin V_i$
  - setze $V_{i+1} = V_i \cup \{y\}$
    $T_{i+1} = T_i \cup \{e\}$
    $i = i + 1$

Lemma 4.3.8

Wenn Algorithmus 4.3.7 endet, dann ist $T = T_i$ aufspannender Baum der Komponente von G, die v enthält.

Minimal aufspannende Bäume

Problem 4.4.3: Minimal aufspannender Baum (MST)

Sei $G = (V, E)$ ein zusammenhängender Graph und sei $w: E \to \mathbb{N}$ eine nichtnegative Kantengewichtsfunktion. Bestimme einen G aufspannenden Baum $T = (V, F)$ minimalen Gewichts $w(F) := \sum_{e \in F} w(e)$.

Algorithmus 4.4.4: Greedy-Algorithmus (Kruskal)

Sortiere die Kanten nach aufsteigendem Kantengewicht $w(e_1) \leq w(e_2) \leq \dots \leq w(e_m)$ und führe Algorithmus 4.3.2 aus.

„Greedy-Algorithmen führen stets den lokal besten nächsten Schritt aus."

Satz 4.4.5

Der Greedy-Algorithmus berechnet einen minimalen aufspannenden Baum.

„Informelle Umformulierung: Streiche solange teuerste Kante, wenn sie auf einem Kreis liegt bis der Graph kreisfrei ist"

Aufgabe 4.4.6 : Kreiskriterium

Sei $G=(V,E)$ ein zusammenhängender Graph, $w: E \to \mathbb{Z}$ eine Kantengewichtsfunktion und $H=(V,T)$ ein G aufspannender Baum.
Dann gilt:

H ist ein minimaler G aufspannender Baum $\iff$
$\forall \bar{e} \in E \setminus T \ \forall e \in C(T,\bar{e}): w(e) \leq w(\bar{e})$.

„In dem von $\bar{e}$ geschlossenen Fundamentalkreis gibt es keine Kante mit höherem Gewicht als $\bar{e}$"

Bemerkung 4.4.7

Voraussetzung: n Zahlen können im Allgemeinen in $O(n \log n)$ Zeit sortiert werden.
Somit Kantensortierung in $O(|E| \log(|E|))$ Zeit.
Mit unserer Implementierung von UNION-FIND erhalten wir für Algorithmus 4.4.4 die Komplexität $O((|E|+|V|) \log(|V|))$.

Algorithmen von Prim-Jarnik und Borůvka

Algorithmus 4.5.1 : Prims Algorithmus

Sei $v \in V$.

- Setze $V_0 = \{v\}$, $T_0 = \emptyset$, $i = 0$
- Solange es geht
  - Wähle Kante $e=(x,y) \in E$ mit $x \in V_i$, $y \notin V_i$ mit minimalem Gewicht
  - Setze $V_{i+1} = V_i \cup \{y\}$
    $T_{i+1} = T_i \cup \{e\}$
    $i = i+1$

„Wähle Startpunkt und füge immer billigste Kante hinzu, so dass der Teilgraph zusammenhängend und kreisfrei bleibt"

Satz 4.5.2

Prims Algorithmus berechnet einen minimal aufspannenden Baum.

„Definition": induzierter Schnitt

Sei $G=(V,E)$ ein zusammenhängender Graph. Ist $S \subseteq V$, so nennen wir die Kantenmenge $\partial_G(S) := \{e \in E \mid |e \cap S| = 1\}$ den von S induzierten Schnitt.
Allgemein nennen wir eine Kantenmenge D einen Schnitt in G, wenn es ein $S \subseteq V$ gibt mit $D = \partial_G(S)$.

„Aufgabe 4.5.3.a"

Sei $G=(V,E)$ ein zusammenhängender Graph und sei $H=(V,T)$ ein G aufspannender Baum. Dann gilt:

$\forall e \in T: \ D(T,e) := \{\bar{e} \in E \mid (T \setminus e) + \bar{e} \text{ ist ein Baum}\}$ ist ein Schnitt in G.

„Aufgabe 5.4.3.b": Schnittkriterium

Sei $G=(V,E)$ ein zusammenhängender Graph und sei $H=(V,T)$ ein G aufspannender Baum. Dann gilt:

H ist ein minimaler G aufspannender Baum $\iff$
$\forall e \in T \ \forall \bar{e} \in D(T,e): w(e) \leq w(\bar{e})$.

Implementierung: Prims Algorithmus

T: speichert Baumkanten
F: Menge der kürzesten Kanten von Knoten in T zu allen Knoten außerhalb von T
F. Minimum Edge (weight): liefert Kante aus F mit minimalem Gewicht
Aktualisieren von F: Prüfe alle ausgehenden Kanten des neuen Baumknotens, ob sie eine kürzere Verbindung zu ihren anderen Endknoten aus T heraus herstellen.

Annahmen:
- Vor Ausführung: $\forall v \in V$: pred[v] = v und weight[(v,v)] = ∞
- Kanten sind gerichtet: Bei (u,v) = F.Maximum Edge (weight) ist u Baumknoten und v ist Knoten außerhalb des Baums

Startknoten v hinzu-
nächste Kante mit T in T aufnehmen

```
T = []   F = []
for w in G.Neighborhood(v):
    F.AddEdge((v,w))
    pred[w] = v
while not T.IsSpanning():
    (u,v) = F.MinimumEdge(weight)
    F.DeleteEdge((u,v))
    T.AddEdge((u,v))
    for w in G.Neighborhood(v):
        if not T.Contains(w) and weight[(pred[w], w)] > weight[(w,v)]:
            F.DeleteEdge((pred[w], w))
            F.AddEdge((v,w))
            pred[w] = v
```

Gesamtlaufzeit $O(|E| \log |V|)$.

„Aufgabe 4.5.5"

Sei $G=(V,E)$ zusammenhängend und sei $w: E \to \mathbb{Z}$ eine Kantengewichtsfunktion. Dann gilt:

a.) $T=(V,F)$ minimaler G aufspannender Baum und $S \subseteq E(T)$
$\Longrightarrow E(T) \setminus S$ Knotenmenge eines minimal aufspannenden Baums von G/S.

b.) Zusätzlich zu Voraussetzungen a.): w ist injektiv
$\Longrightarrow$ Menge S der Kanten kleinsten Gewichts an jedem Knoten ist in dem eindeutigen minimalen aufspannenden Baum enthalten.

Algorithmus 4.5.6: Borůvkas Algorithmus

Sei $G=(V,E)$ ein Multigraph mit injektiver Gewichtsfunktion.
Setze $T = \emptyset$.
Solange G mehr als einen Knoten hat:
  Markiere zu jedem Knoten die Kante minimalen Gewichts, die zu ihm adjazent und keine Schleife ist.
  Füge alle markierten Kanten S zu T hinzu und setze $G = G/S$.

Satz 4.5.7

Borůvkas Algorithmus berechnet den eindeutigen minimalen aufspannenden Baum von G.

Bemerkung 4.5.8

Injektivität von w ist keine wirkliche Einschränkung, da bei mehrfach vorkommenden Kantengewichten z.B. mithilfe der Kantennummern eine „echte Ungleichung" herzustellen.

Gesamtlaufzeit $O(|E| \log |V|)$

„Aufgabe 4.5.10": Teilbarkeitsgraph

Wir nennen $G=(V,E)$ definiert durch $e=(i,j) \in E \iff (i \mid j$ oder $j \mid i)$ und $i \neq j$ einen Teilbarkeitsgraphen.

Anzahl aufspannender Bäume

„Definition": knotengelabelte Bäume

Wir betrachten isomorphe, aber nicht identische Bäume als verschieden und nennen diese knotengelabelte Bäume.

„Definition": Knoten- und kantengelabelte Wurzelbäume

Ein Knoten- und kantengelabelter Wurzelbaum ist ein Wurzelbaum mit einer Kantennummerierung mit Werten in $\{1, \dots, n-1\}$.

Proposition 4.6.1

Jeder knotengelabelte Baum mit n Knoten gibt Anlass zu genau $n!$ knoten- und kantengelabelten Wurzelbäumen.

Lemma 4.6.2

Es gibt genau $n! \, n^{n-2}$ knoten- und kantengelabelte Wurzelbäume mit n Knoten.

Satz 4.6.3: Cayley-Formel

Die Anzahl der knotengelabelten Bäume mit n Knoten ist n^{n-2}.

Bipartites Matching

Definition 4.7.2: bipartiter Graph

Sei $G=(W,E)$ ein Graph. Wir nennen G bipartit, wenn es eine Partition $W = U \dot{\cup} V$ gibt, so dass alle Kanten je einen Endknoten in beiden Klassen haben. Wir nennen U und V dann die Farbklassen von G.

Proposition 4.7.3

Ein Graph $G=(W,E)$ ist genau dann bipartit, wenn er keinen Kreis ungerader Länge hat.

Definition 4.7.4: Matching

Sei $G=(V,E)$ ein Graph. Wir nennen eine Kantenmenge $M \subseteq E$ ein Matching in G, wenn für den Graphen $G_M=(V,M)$ gilt

$\forall v \in V: \deg_{G_M}(v) \leq 1.$

Ist $(u,v) \in M$, so sagen wir, dass u mit v gematched ist. Ist ein Knoten mit einer Matchingkante inzident, so nennen wir den Knoten gematched, ansonsten ungematched.

Wir bezeichnen obige Bedingung auch als Bigamieverbot. Gilt bezüglich der Bedingung stets Gleichheit, so sprechen wir von einem perfekten Matching.

„Definition": Symmetrische Differenz

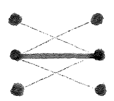
Seien A, B Mengen. Wir definieren die symmetrischen Differenz

$A \Delta B = (A \setminus B) \cup (B \setminus A).$

„Definition": inklusionsmaximales Matching

Kann einem Matching keine weitere Kante hinzugefügt werden ohne das Bigamieverbot zu verletzen, so nennen wir das Matching inklusionmaximal.

Beispiel: inklusionsmaximales Matching (nur fette Kante)



Definition 4.7.5: alternierender / augmentierender Weg

Sei $G=(V,E)$ ein (nicht notwendigerweise bipartiter) Graph und $M \subseteq E$ ein Matching. Sei $P = v_0 v_1 v_2 \cdots v_k$ ein Weg. Wir sagen

a.) Weg P ist M-alternierend, wenn seine Kanten abwechselnd in M und außerhalb von M liegen.

b.) Weg P ist M-augmentierend, wenn er M-alternierend ist und die Randknoten v_0 und v_k ungematched sind.
Insbesondere ist dann k ungerade und
$v_i v_{i+1} \in M \iff 1 \leq i \leq k-2$ und i ungerade

Satz 4.7.6

Sei $G=(V,E)$ ein (nicht notwendig bipartiter Graph) und $M \subseteq E$ ein Matching. Dann gilt:

M ist von maximaler Kardinalität $\iff$
es gibt keinen M-augmentierenden Weg in G.

Proposition 4.7.7

Sei $G=(U \,\dot\cup\, V, E)$ ein bipartiter Graph, M ein Matching in G und $P = v_0 v_1 v_2 \cdots v_k$ ein M-augmentierender Weg in G. Dann gilt:

$v_0 \in U \iff v_k \in V.$

Algorithmus 4.7.8: Find-Augmenting-Path

Input: bipartiter Graph $G=(U \,\dot\cup\, V, E)$ und Matching $M \subseteq E$
Output: Endknoten eines M-augmentierenden Weges oder „Maximalzertifikat" C

Queue Q: enthält noch zu bearbeitende Knoten (FiFo)
Initialisierung: pred[v] = None für alle $v \in V$, C ist leere Liste

```
for u in U:
    if not M.IsMatched(u):        } alle ungematchten Knoten aus U in Q sammeln
        Q.Append(u)
        pred[u] = u
while not Q.IsEmpty():
    u = Q.Top()
    for v in G.Neighborhood(u):
        if pred[v] = None:        } Wenn betrachtete Nachbar
            pred[v] = u             a) nicht gematched -> augmentierender Weg
            if not M.IsMatched(v):     gefunden, Abbruch
                return v
            else:                   b) ist gematched -> Matchingpartner in
                s = M.Partner(v)       Q aufnehmen
                Q.Append(s)
                pred[s] = v
for u in U:                       } Falls in Q kein augmentierender Weg
    if pred[u] == None:             gefunden wurde:
        C.Append(u)                 Ausgabe des „Maximalzertifikats" C
for v in V:
    if pred[v] != None:
        C.Append(v)
```

Proposition 4.7.10

Findet die Prozedur „Find Augmenting Path" einen ungematchten Knoten v, so erhält man durch Rückverfolgen des pred-Arrays einen M-augmentierenden Weg.

Definition 4.7.11: Kantenüberdeckende Knotenmenge / Knotenüberdeckung

Sei $G=(V,E)$ ein (nicht notwendig bipartiter) Graph und $C \subseteq V$. Wir nennen C kantenüberdeckende Knotenmenge oder Knotenüberdeckung (engl. vertex cover), wenn gilt:

$\forall e \in E: C \cap e \neq \emptyset$

Lemma 4.7.12

Liefert das Verfahren eine Knotenliste C zurück, so ist $|C| = |M|$ und C ist eine kantenüberdeckende Knotenmenge.

Satz 4.7.13: Satz von König (1931)

In bipartiten Graphen ist
$\max\{|M| \mid M \text{ ist Matching}\} = \min\{|C| \mid C \text{ ist Knotenüberdeckung}\}$

Bemerkung 4.7.14

In allgemeinen Graphen ist das Problem der minimalen Knotenüberdeckung NP-vollständig.

Definition 4.7.15: Nachbarschaft

Ist $G=(V,E)$ ein (nicht notwendig bipartiter) Graph und $H \subseteq V$, so bezeichnen wir mit $N_G(H)$ bzw. $N(H)$ die Nachbarschaft von H

$N(H) := \{v \in V \mid \exists u \in H: (u,v) \in E\}.$

Korollar 4.7.16: Heiratssatz von Frobenius (1917)

Sei $G=(U \,\dot\cup\, V, E)$ ein bipartiter Graph. Dann gilt:

G hat ein perfektes Matching $\iff |U| = |V|$ und $\forall H \subseteq U: |N(H)| \geq |H|.$

Satz 4.7.17: Heiratssatz von Hall

Sei $G=(M \,\dot\cup\, V, E)$ ein bipartiter Graph. Dann gilt:

G hat ein Matching, in dem alle Knoten in M gematched sind $\iff$
$\iff \forall H \subseteq M: |N(H)| \geq |H|.$

Algorithmus 4.7.19: Bipartites Matching

Starte mit einem leeren Matching M und setze $C = [\,]$.

```
while C == []:
    (C, w) = FindAugmentingPath(M)
    if C == []:
        P = BackTrackPath(w)
        Augment(M, P)
```

Gesamtlaufzeit $O(\min\{|U|,|V|\}|E|)$

„Definition": Permutationsmatrix

Wir nennen eine Matrix $P \in \{0,1\}^{n \times n}$, bei der in jeder Zeile und Spalte jeweils genau eine 1 und sonst nur Nullen stehen, eine Permutationsmatrix.

„Definition": doppelt stochastisch

Wir nennen eine Matrix $A \in \mathbb{R}^{n \times n}$, bei der alle Einträge $0 \leq a_{ij} \leq 1$ sind, doppelt stochastisch, wenn die Summe aller Einträge in jeder Zeile und Spalte gleich ist.

„Aufgabe 4.7.23": Konvexkombination von Permutationsmatrizen

Jede doppelt statistische Matrix ist eine Konvexkombination von Permutationsmatrizen, d.h. es gibt

$L \in \mathbb{N}$ und Permutationsmatrizen $P_1, \dots, P_L$ und Koeffizienten $\lambda_1, \dots, \lambda_L$ mit $0 \leq \lambda_i \leq 1$ und $\sum_{i=1}^{L} \lambda_i = 1$,

so dass $A = \sum_{i=1}^{L} \lambda_i P_i$.

Stabile Hochzeiten

Problem 4.8.1

In einer geschlossenen Gesellschaft gibt es je n heiratsfähige Männer und Frauen. Sowohl Männer als auch Frauen haben Präferenzen, was die Personen des anderen Geschlechts angeht. Verheirate Männer und Frauen so, dass es kein Paar aus Mann und Frau gibt, die nicht miteinander verheiratet sind, sich aber gegenseitig ihren Ehepartnern vorziehen.

Definition 4.8.2: stabile Hochzeit

Seien M, V Mengen mit $|M| = |V|$. Sei für alle $u \in M$ $\prec_u$ eine Totalordnung von V und für alle $v \in V$ $\prec_v$ eine Totalordnung von V. Wir nennen eine bijektive Abbildung $\tau: M \to V$ eine stabile Hochzeit, wenn

$$\forall u \in M\ \forall v \in V: \begin{cases} \tau(u) = v & \text{oder} \\ v \prec_u \tau(u) & \text{oder} \\ u \prec_v \tau^{-1}(v) & \end{cases} \quad \text{gilt.}$$

„Entweder sind u und v miteinander verheiratet oder mindestens einer zieht seinen Ehepartner dem anderen (d.h. u oder v) vor."

Algorithmus 4.8.3: Men propose - Women dispose

Eingabedaten wie eben. Zu Anfang ist niemand verlobt. Die verlobten Paare bilden stets ein Matching im vollständigen bipartiten Graphen auf M und V. Der Algorithmus terminiert, wenn das Matching perfekt ist.

- Solange es einen Mann gibt, der noch nicht verlobt ist, macht dieser der besten Frau auf seiner Liste einen Antrag.
- Wenn die Frau nicht verlobt ist oder ihr der Antragsteller besser gefällt als ihr Verlobter, nimmt sie den Antrag an und löst ggf. ihre alte Verlobung. Ihr Ex-Verlobter streicht sie von seiner Liste.
- Andernfalls lehnt sie den Antrag ab und der Antragsteller streicht sie von seiner Liste.

Proposition 4.8.4

Kein Mann macht der gleichen Frau zweimal einen Antrag.

Proposition 4.8.5

Eine Frau, die einmal verlobt ist, bleibt es und wird zu keinem späteren Zeitpunkt mit einem Antragsteller verlobt sein, der ihr schlechter als ihr derzeitiger Verlobter gefällt.

Satz 4.8.6

a.) Der Algorithmus „Men propose - Women dispose" terminiert in $O(n^2)$.

b.) Wenn er terminiert, sind alle verlobt.

c.) Die durch die Verlobungen definierte bijektive Abbildung ist eine stabile Hochzeit.

„Aufgabe 4.8.7": männeroptimale stabile Hochzeit

Sei $u \in M$ ein beliebiger Mann und τ das Ergebnis von „Men propose - Woman dispose". Gibt es keine stabile Hochzeit σ, in der u mit einer Frau verheiratet wird, die er seiner gegenwärtigen vorzieht, also für jede stabile Hochzeit σ gilt:

$$\forall u \in M: \sigma(u) \preceq_u \tau(u),$$

so sprechen wir von einer männeroptimalen stabilen Hochzeit.

Notation

Intervalle

Seien $a, b \in \mathbb{R}$. Wir nennen die Menge
$[a,b] := \{x \in \mathbb{R} \mid a \le x \le b\}$ das abgeschlossene Intervall,
$]a,b[:= \{x \in \mathbb{R} \mid a < x < b\}$ das offene Intervall,
$[a,b[:= \{x \in \mathbb{R} \mid a \le x < b\}$ das rechts halboffene Intervall,
$]a,b] := \{x \in \mathbb{R} \mid a < x \le b\}$ das links halboffene Intervall.

„Definition": Menge aller Abbildungen

Seien E und R Mengen. Wir bezeichnen mit der Menge $R^E := \{f : E \to R\}$ die Menge aller Abbildungen von E nach R.

Ist E endlich, so betrachten wir die Elemente R^E auch als $|E|$-Vektoren und schreiben $x = (x_e)_{e \in E}$.

„Definitionen": Vektoren

Vektorenmenge
Für $n \in \mathbb{N}$ bezeichnet $\mathbb{N}^n$ ($\mathbb{Z}^n$, $\mathbb{Q}^n$, $\mathbb{R}^n$) die Menge der Vektoren mit n Komponenten mit Einträgen in $\mathbb{N}$ ($\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$).

Spaltenvektoren
Soweit nicht explizit anders gesagt, sind Vektoren stets Spaltenvektoren.

Transposition
Die Transposition x^T eines Spaltenvektors x ist ein Zeilenvektor.

Skalarprodukt von x und y
Das Matrixprodukt $x^T y = \sum_{i=1}^{n} x_i y_i$ von $x, y \in \mathbb{R}^n$ ist das Skalarprodukt von x und y.

Vektorvergleich
Seien $a, b \in \mathbb{R}^n$. Dann gilt:
$a \le b \iff \forall i = 1, \dots, n : a_i \le b_i$.

Menge (skalar?)
Sei $M \subseteq \mathbb{R}^n$ und $\alpha \in \mathbb{R}$, so gilt $\alpha M := \{\alpha x \mid x \in M\}$.

orthogonales Komplement von M
Die Menge $M^\perp := \{y \in \mathbb{R}^n \mid \forall x \in M : x^T y = 0\}$ wird als das orthogonale Komplement von M bezeichnet.

i-ter Einheitsvektor
Wir bezeichnen mit $e_i \in \mathbb{R}^n$ den i-ten Einheitsvektor, also $(e_i)_j = 1$, falls $i = j$ und sonst 0.

Normen

Sei $x \in \mathbb{R}^n$.

L_1 oder 1-Norm

$$\|x\|_1 := \sum_{i=1}^{n} |x_i|$$

L_2-Norm / euklidische Norm

$$\|x\|_2 := \sqrt{x^T x} = \sqrt{\sum_{i=1}^{n} x_i^2}$$

! Sofern nicht ausdrücklich anders definiert, bezeichnet $\|x\|$ stets die euklidische Norm!

L_∞-Norm / Maximumnorm

$$\|x\|_\infty := \max_{1 \le i \le n} |x_i|$$

offene ε-Umgebung

Ist $P \subseteq \mathbb{R}^n$ und $\varepsilon > 0$, so bezeichnen wir mit
$M_\varepsilon(P) := \{x \in \mathbb{R}^n \mid \exists p \in P : \|x - p\| < \varepsilon\}$
die offene ε-Umgebung der Menge P.
Ist $P = \{p\}$ einelementig, so schreiben wir statt $M_\varepsilon(\{p\})$ kurz $M_\varepsilon(p)$.

Matrizen

Matrizenmenge

Wir bezeichnen mit $\mathbb{R}^{m \times n}$ die Menge der $(m \times n)$-Matrizen mit Einträgen in $\mathbb{R}$. Dabei gibt m die Zeilen- und n die Spaltenanzahl an. Einträge werden mit a_{ij} oder A_{ij} mit $1 \le i \le m$ und $1 \le j \le n$ adressiert.

„Matrixeinschränkung"

Seien M und N die Zeilen- bzw. Spaltenmenge einer Matrix A und seien $I \subseteq M$ und $J \subseteq N$. Wir bezeichnen mit A_{IJ} die Matrix $A_{IJ} = (a_{ij})_{i \in I, j \in J}$.
Statt A_{IN} schreiben wir $A_{I.}$ und statt A_{MJ} $A_{.J}$.

Einheitsmatrix

Die Einheitsmatrix I_n ist eine $(n \times n)$-Matrix mit Einsen auf der Diagonalen und sonst nur Nullen.

Transponierte

Sei $A = (a_{ij})$ eine $(m \times n)$-Matrix. Dann ist die $(n \times m)$-Matrix $A^T = (a_{ji})$ die Transponierte von A. „An Diagonalen spiegeln" „Erste Zeile wird zu erste Spalte etc."

Symmetrische Matrix

Eine quadratische Matrix A heißt symmetrisch, wenn $A^T = A$ ist. „An Diagonalen symmetrisch"

Modulo

Seien $A, B \in \mathbb{N}$ mit $B \ne 0$. Wir bezeichnen den Rest der ganzzahligen Division von A durch B als $A \bmod B$.
Es gilt $A \bmod B = A - \left\lfloor \frac{A}{B} \right\rfloor B$.

Kodierung von Zahlen

Satz 5.2.1

Sei $B \in \mathbb{N}$ mit $B \ge 2$ und sei $x \in \mathbb{R} \setminus \{0\}$. Dann gibt es genau eine Darstellung der Gestalt

$$x = \sigma B^n \sum_{i=1}^{\infty} x_{-i} B^{-i} \quad \text{mit}$$

a.) $\sigma \in \{+1, -1\}$ „Vorzeichen von x"
b.) $n \in \mathbb{Z}$
c.) $x_{-i} \in \{0, 1, \dots, B-1\}$
d.) $x_{-1} \ne 0$ und zusätzlich
e.) $\forall j \in \mathbb{N}\ \exists k \ge j : x_{-k} \ne B-1$. „Schließt Perioden aus"

Beispiele

- Basis 10

$$\begin{aligned}
&123456.789 \\
&= 10^6 (1 \cdot 10^{-1} + 2 \cdot 10^{-2} + 3 \cdot 10^{-3} + 4 \cdot 10^{-4} + 5 \cdot 10^{-5} + 6 \cdot 10^{-6} + 7 \cdot 10^{-7} + 8 \cdot 10^{-8} + 9 \cdot 10^{-9}) \\
&= 1 \cdot 10^5 + 2 \cdot 10^4 + 3 \cdot 10^3 + 4 \cdot 10^2 + 5 \cdot 10^1 + 6 \cdot 10^0 + 7 \cdot 10^{-1} + 8 \cdot 10^{-2} + 9 \cdot 10^{-3}
\end{aligned}$$

Bemerkung 5.2.2

Für allgemeine $B \in \mathbb{N}$ mit $B \ge 2$ heißt die Darstellung aus Satz 5.2.1 die B-adische Darstellung von x.

Im Rechner verwenden wir $B \in \{2, 8, 16\}$. Die Darstellungen begegnen uns als Binär-, Oktal- und Hexadezimalzahlen.

Fehlerquellen

„Definition": absoluter und reeller Fehler

Sei $x \in \mathbb{R}$ und sei $\tilde{x} \in \mathbb{R}$ eine Näherung für x. Dann heißt:

a.) $x - \tilde{x}$ der absolute Fehler und

b.) für $x \neq 0$ $\frac{x-\tilde{x}}{x}$ der relative Fehler.

„Vereinbarung": Rundungsvorschrift „Kaufmännisches Runden"

Bei der Darstellung von Zahlen bzgl. einer Basis $B \in \mathbb{N}$ mit $B \geq 2$ wollen wir folgende Rundungsvorschrift für die t-stellige Darstellung mit $x \in \mathbb{R} \setminus \{0\}$ für

$x = \sigma B^n \sum_{i=1}^{\infty} x_{-i} B^{-i}$ benutzen:

$$Rd_t(x) = \begin{cases} \sigma B^n \sum_{i=1}^{t} x_{-i} B^{-i}, & \text{falls } x_{-t-1} < \frac{B}{2} \\ \sigma B^n \left(B^{-t} + \sum_{i=1}^{t} x_{-i} B^{-i}\right), & \text{falls } x_{-t-1} \geq \frac{B}{2} \end{cases}$$

„Definition": Maschinenzahl

Sei nun eine Maschinengenauigkeit $t \in \mathbb{N}$ vorgegeben. Dann bezeichnen wir Zahlen mit einer Darstellung

$$\sigma B^n \sum_{i=1}^{t} x_{-i} B^{-i}$$

als Maschinen- oder Gleitkommazahlen.

$$\underset{\text{Vorzeichen}}{\sigma}\ \underset{\text{Exponent}}{B^n}\ \underbrace{\sum_{i=1}^{t} x_{-i} B^{-i}}_{\text{t-stellige Mantisse}}.$$

Satz 5.3.3

Seien $B \in \mathbb{N}$ mit $B = 2k \geq 2$, $t \in \mathbb{N}$ und $x = \sigma B^n \sum_{i=1}^{\infty} x_{-i} B^{-i} \neq 0$. Dann gilt:

a.) $Rd_t(x)$ hat eine Darstellung der Gestalt $\sigma B^{n'} \sum_{i=1}^{t} x'_{-i} B^{-i} \neq 0$

b.) der absolute Fehler ist beschränkt durch $|x - Rd_t(x)| < \frac{B^{n-t}}{2}$

c.) der relative Fehler ist beschränkt durch $\left|\frac{x - Rd_t(x)}{x}\right| < \frac{B^{1-t}}{2}$

d.) der relative Fehler bzgl. $Rd_t(x)$ ist beschränkt durch $\left|\frac{x - Rd_t(x)}{Rd_t(x)}\right| \leq \frac{B^{1-t}}{2}$.

Gaußelimination und Pivotstrategien

Lösen linearer Gleichungssysteme

Seien $A \in \mathbb{R}^{m \times n}$ und $b \in \mathbb{R}^m$. Gesucht wird ein $x \in \mathbb{R}^n$, so dass $Ax = b$ gilt.

Algorithmentypen zum Lösen linearer Gleichungssysteme

a.) direkte Verfahren: errechnen in endlich vielen Schritten eine mit Rundungsfehlern behaftete Lösung (wird hier betrachtet)

b.) indirekte, iterative Verfahren: brechen beim Erreichen einer gewissen Genauigkeit ab

Vorgehen zum Lösen

Wähle zeilenweise Pivotelemente (vorzugsweise auf der Diagonalen) aus und bringe die Matrix durch elementare Zeilenumformungen in Zeilenstufenform (unterhalb der Diagonalen nur Nullen).

Gaußeliminationen

Wähle ein Pivotelement aus und sorge mit elementaren Zeilenumformungen dafür, dass unterhalb des Pivotelements nur Nullen stehen.

Sei $a_{ij} \neq 0$ das Pivotelement. Für die Gaußelimination ziehen wir für alle Indizes $k > i$ das $\frac{a_{kj}}{a_{ij}}$-fache der i-ten Zeile von der k-ten ab. Für die neue k-te Zeile $\tilde{A}_{k.}$ gilt somit $\tilde{A}_{k.} = A_{k.} - \frac{a_{kj}}{a_{ij}} A_{i.}$

Algorithmus: Gaußelimination

```
def gausselim(A, i, j):
    for k in range(i+1, m):
        A_kj = A[k,j]
        A_ij = A[i,j]
        Q_kj = - A_kj / A_ij
        A[k,:] = A[k,:] + Q_kj * A[i,:]
```

Zeilenweiser Zugriff auf Matrix A

Behandlung von Nullstellen auf der Diagonalen

Wenn es bei der Bestimmung des Pivotelementes auf der Diagonalen die Komponente $a_{jj} = 0$ ist und in einer späteren oder gleichen Zeile und in einer späteren oder gleichen Spalte ein $a_{kl} \neq 0$ mit $k \geq j$ und $l \geq j$ gibt, dann:

- vertausche k-te und j-te Zeile
- vertausche l-te und j-te Spalte
- führe Gaußelimination auf a_{jj} aus
- merke: in Lösung Indizes l und j vertauschen (Spalten)

Algorithmus: Gaußalgorithmus

```
def gaussalg(A, b):
    d = min([m, n])
    index = range(0, n)              Permutation der Variablen
    C = concatenate((A, b), 1)       Verschmelze A, b nebeneinander
    for k in range(d):
        if C[k,k] == 0:
            pivotfound, i, j = findpivot(C, index, k)
            if pivotfound == 0:
                break
            else:
                if i != k:
                    swaprows(C, i, k)
                if j != k:
                    swapcolumns(C, index, j, k)     Vertausche Spalte und Variablen j und k
        gausselim(C, k, k)
    solvable, x = compute_x(C, index, k)
    return solvable, x
```

Prüfe, ob LGS lösbar und bestimme Lösung

Ziel ist es, anhand der transformierten Matrix festzustellen, ob das LGS lösbar ist und ggf. eine spezielle Lösung zu bestimmen.

Vorgehen ohne Spaltenvertauschung:

- setze $x_m = \frac{b_m}{c_{mm}}$
- iteriere rückwärts
  $x_i = \frac{b_i - \sum_{j=i+1}^{m} c_{ij} x_j}{c_{ii}}$

Algorithmus: compute_x

Hier Anfang weggelassen: prüfe transformiertes Gleichungssystem auf Nullzeilen oder Unlösbarkeit

```
def compute_x(C, ind, d):
    # C = [A|b] mit A ist obere Dreiecksmatrix mit m <= n
    # Wir lösen das System Ax=b unter Berücksichtigung
      index für x.
    if solvable:
        x[ind[d]] = C[d,n] / C[d,d]
        for i in range(1, d+1):
            x[ind[d-i]] = C[d-i,n]
            for j in range(0,i):
                x[ind[d-i]] = x[ind[d-i]] - C[d-i, d-j]
                              * x[ind[d-j]]
            x[ind[d-i]] = x[ind[d-i]] / C[d-i, d-i]
    return solvable, x
```

Bemerkung 5.4.2

Grundsätzlich sollte bei numerischen Berechnungen wegen Rundungsfehlern Abfragen auf Gleichheit vermeiden. Besser $|x| \geq \varepsilon$ mit kleiner Konstante abfragen.

LU-Zerlegung

Definition 5.5.1: untere / obere Dreiecksmatrix

Sei A eine $(n\times n)$-Matrix. Wir nennen A

a.) obere Dreiecksmatrix, wenn unterhalb der Diagonalen nur Nullen stehen, also $A_{ij}=0$ für alle $i>j$ und $1\le i,j\le n$.

b.) untere Dreiecksmatrix, wenn oberhalb der Diagonalen nur Nullen stehen, also $A_{ij}=0$ für alle $i<j$ und $1\le i,j\le n$.

„Definition": reguläre Matrix

Wir bezeichnen eine Matrix mit linear unabhängigen Zeilen und Spalten als reguläre Matrix.

Proposition 5.5.2

Sei A eine reguläre $(n\times n)$-Matrix und sei $1\le d\le n>0$, dann für alle $0<j<d$ und alle $i>j$ die Bedingung $A_{ij}=0$ erfüllt ist. (A hat also in den ersten $d-1$ Spalten die Gestalt einer oberen Dreiecksmatrix)

Dann gibt es ein i mit $d\le i\le n$ mit $A_{id}\neq 0$.

„Hat die reguläre Matrix A in den ersten $d-1$ Spalten die Gestalt einer oberen Dreiecksmatrix, so gibt es in Spalte d beginnend ab dem Diagonalelement mindestens einen Eintrag $\neq 0$."

Definition 5.5.3: Transpositions- und Permutationsmatrix

Sei $i\neq j$. Eine $(n\times n)$-Matrix der Gestalt

$$P^n_{ij} = \begin{array}{c} \\ \\ i \\ \\ j \\ \\ \\ \end{array}\!\!\begin{array}{c} \begin{array}{ccccccc} & & i & & j & & \end{array} \\ \begin{pmatrix} 1 & & & & & & \\ & 1 & & & 0 & & \\ & & 0 & \cdots & 1 & & \\ & & \vdots & 1 & \vdots & & \\ & & 1 & \cdots & 0 & & \\ & 0 & & & & 1 & \\ & & & & & & 1 \end{pmatrix}\end{array}$$

also eine symmetrische Matrix mit n Einsen, von denen $n-2$ auf der Diagonalen stehen, und sonst nur Nulleinträge, heißt Transpositionsmatrix.

Ein Produkt von Transpositionsmatrizen nennen wir eine Permutationsmatrix.

Bemerkung 5.5.4

Sei A eine $(m\times n)$-Matrix.

- Multiplikation $P^m_{ij}\cdot A$ vertauscht i-te mit der j-ten Zeile
- Multiplikation $A\cdot P^n_{ij}$ vertauscht i-te mit der j-ten Spalte

Proposition 5.5.5

a.) $P^n_{ij} = I_n - e_ie_i^T - e_je_j^T + e_ie_j^T + e_je_i^T$

b.) Transpositionsmatrizen sind selbstinvers, d.h. $P^n_{ij}P^n_{ij} = I_n$.

Beachte:

- Produkt $e_ie_j^T$ eines Spaltenvektors mit einem Zeilenvektor ist eine $(n\times n)$-Matrix.
- Produkt eines Zeilenvektors mit einem Spaltenvektor ist ein Skalarprodukt.

Definition 5.5.8: Frobeniusmatrix

Eine $(n\times n)$-Matrix der Gestalt

$$G_d = \begin{array}{c} \\ \\ d \\ \\ \\ \\ \end{array}\!\!\begin{array}{c} \begin{array}{cccccc} & & d & & & \end{array} \\ \begin{vmatrix} 1 & \cdots & 0 & & \cdots & 0 \\ \vdots & \ddots & \vdots & & \ddots & \vdots \\ 0 & \cdots & 1 & & \cdots & 0 \\ 0 & \cdots & -\ell_{d+1,d} & & \cdots & 0 \\ \vdots & \ddots & \vdots & & \ddots & 0 \\ 0 & \cdots & -\ell_{n,d} & & \cdots & 1 \end{vmatrix}\end{array}$$

heißt Frobeniusmatrix. Sie unterscheidet sich höchstens in der d-ten Spalte von einer Einheitsmatrix.

Wir können also schreiben

$$G_d = I_n - \ell_d e_d^T \quad \text{mit}$$

$$\ell_d = (0,\dots,0,\ell_{d+1,d},\dots,\ell_{n,d})^T.$$

Bemerkung 5.5.9

Die Frobeniusmatrix ist regulär und man rechnet nach:

$$G_d^{-1} = I_n + \ell_d e_d^T = \begin{array}{c} \\ \\ d \\ \\ \\ \\ \end{array}\!\!\begin{array}{c} \begin{array}{cccccc} & & d & & & \end{array} \\ \begin{pmatrix} 1 & \cdots & 0 & & \cdots & 0 \\ \vdots & \ddots & \vdots & & \ddots & \vdots \\ 0 & \cdots & 1 & & \cdots & 0 \\ 0 & \cdots & \ell_{d+1,d} & & \cdots & 0 \\ \vdots & \ddots & \vdots & & \ddots & 0 \\ 0 & \cdots & \ell_{n,d} & & \cdots & 1 \end{pmatrix}\end{array}.$$

Denn $(I_n-\ell_de_d^T)(I_n+\ell_de_d^T)$

$$= I_n + \ell_de_d^T - \ell_de_d^T - (\ell_de_d^T)(\ell_de_d^T)$$

$$= I_n - \ell_d(e_d^T\ell_d)e_d^T = I_n, \quad \text{denn } e_d^T\ell_d = (\ell_d)_d = 0.$$

„Gaußeliminierung durch Frobeniusmatrix"

Sei a_{dd} ein Pivotelement der Matrix A. Der Gaußeliminationsschritt bezüglich a_{dd} wird durch Linksmultiplikation einer Frobeniusmatrix mit $\ell_{id} = \dfrac{A_{id}}{A_{dd}}$ bewirkt.

„Ziel"

Aggregation der bei der Gaußelimination benötigten Transpositionsmatrizen in einer einzigen Permutationsmatrix.

Bei uns wechseln sich die Transpositions- mit Frobeniusmatrizen ab
→ Transpositionsmatrizen an den Frobeniusmatrizen „vorbeiziehen"

Proposition 5.5.10

a.) Ist P^n_{ij} eine Transpositionsmatrix mit $i,j>d$, so gilt

$$P^n_{ij}G_d^{-1}P^n_{ij} = P^n_{ij}(I_n+\ell_de_d^T)P^n_{ij} = I_n + (P^n_{ij}\ell_d)e_d^T.$$

b.) Sind $P^n_{i_1j_1},\dots,P^n_{i_kj_k}$ Transpositionsmatrizen mit $i_l,j_l>d$, so gilt

$$\prod_{l=1}^{k} P^n_{i_lj_l}\,G_d^{-1}\prod_{l=1}^{k}P_{i_{k+1-l}j_{k+1-l}} = I_n + \Big(\big(\prod_{l=1}^{k}P^n_{i_lj_l}\big)\ell_d\Big)e_d^T.$$

Satz 5.5.11: Satz von der Dreieckszerlegung

Sei $A\in\mathbb{R}^{n\times n}$ eine reguläre Matrix und seien $P^n_{i_1j_1},\dots,P^n_{i_{n-1}j_{n-1}}$ und $G_1,\dots,G_{n-1}$ die benötigten Transpositions- bzw. Frobeniusmatrizen.

Setzen wir

$$P = \prod_{k=1}^{n-1}P^n_{i_{n-k}j_{n-k}} \quad \text{und} \quad L = \prod_{k=1}^{n-1}\Big(I_n + \Big(\prod_{s=1}^{n-k-1}P^n_{i_{n-s}j_{n-s}}\Big)\ell_ke_k^T\Big)$$

und bezeichnen mit M die Transformierte von A, also die obere Dreiecksmatrix, die das „Ergebnis" unserer Transformationen ist.

Dann ist L eine untere Dreiecksmatrix und es gilt

$$PA = LU$$

Lemma 5.5.12

Sei für $1\le r\le n-1$ die Matrix $L=\prod_{k=1}^{r}(I_n+\ell_ke_k^T)$ Produkt von Frobeniusmatrizen mit wachsendem Index. Dann ist

$L = I_n + \sum_{k=1}^{r}\ell_ke_k^T$, also eine untere Dreiecksmatrix mit nur Einsen auf der Diagonalen, und die Spalten $r+1,\dots,n$ sind Einheitsvektoren.

Bemerkung 5.5.13

Die LU-Zerlegung ist nützlich, um $Ax=b$ für verschiedene Vektoren b lösen muss. Dazu berechnet man zunächst ein $c\in\mathbb{R}^n$ mit $Lc=Pb$. Da L eine untere Dreiecksmatrix ist, lässt sich diese Gleichung durch eine einfache Rekursion lösen. Nun berechnet man ebenso einfach ein $x\in\mathbb{R}^n$ mit $Mx=c$. Insgesamt gilt dann

$$Ax = P^{-1}PAx = P^{-1}LUx = P^{-1}Lc = P^{-1}Pb = b.$$

Gauß-Jordan-Algorithmus

„Gauß-Jordan-Elimination"

Eliminiere Variablen über und unterhalb der Pivotposition.

Algorithmus: Gauß-Jordan-Elimination

```
def gaussjordanelim (A,i,j):
    A_ij = A[i,j]
    A[i,:] = A[i,:]/A_ij
    for k in range(i) + range (i+1,m):
        A_kj = A[k,j]
        A[k,:] = A[k,:] - A[i,:]* A_kj
```

„range (i) + range (i+1, m) erzeugt eine Liste von 0 bis m-1, die das Element i auslässt."

„Gauß-Jordan-Algorithmus"

Eliminationsschritte können zu einem Algorithmus kombiniert werden, bei dem in A eine Einheitsmatrix übrigbleibt und rechts eine Lösung des linearen Gleichungssystems steht.

„Eliminationsschritt durch Linksmultiplikation einer Matrix"

Sei a_{ij} das Pivotelement. Dann kann der Gauß-Jordan-Eliminationsschritt als Linksmultiplikation der Matrix

$$G_{i,j} = \begin{array}{c} \\ \\ \\ i \\ \\ \\ \\ \end{array} \begin{array}{c} \begin{array}{ccccc} & & i & & \end{array} \\ \begin{pmatrix} 1 & & -\frac{a_{1j}}{a_{ij}} & & \\ & \ddots & \vdots & & 0 \\ & & \frac{1}{a_{ij}} & & \\ & & -\frac{a_{i+1,j}}{a_{ij}} & \ddots & \\ 0 & & \vdots & & \\ & & -\frac{a_{nj}}{a_{ij}} & & 1 \end{pmatrix} \end{array}.$$

Wir bezeichnen diese Matrix als η-Matrix und nennen die i-te Spalte η-Vektor.

Elementares über Eigenwerte

Definition 5.7.1: Eigenwert

Sei $A \in \mathbb{R}^{n \times n}$. Wir nennen $\lambda \in \mathbb{R}$ Eigenwert von A, wenn ein $x \in \mathbb{R}^n \setminus \{0\}$ mit $Ax = \lambda x$ existiert.

Jeder solche Vektor $x \neq 0$ heißt Eigenvektor von A zum Eigenwert λ.

„Aufgabe 5.7.2"

Sei $A \in \mathbb{R}^{n \times n}$ und $\lambda \in \mathbb{R}$. Dann gilt:

λ ist Eigenwert von A $\iff$ $(A - \lambda I_n)$ ist eine nicht reguläre Matrix.

Definition: orthogonal

Eine quadratische Matrix $Q \in \mathbb{R}^{n \times n}$ heißt orthogonal, wenn $Q^T Q = Q Q^T = I_n$ gilt.

„Bemerkung"

Orthogonale Matrizen beschreiben bzgl. der Standardbasen des $\mathbb{R}^n$ lineare Abbildungen, die Länge und Winkel konstant lassen.

Satz 5.7.4: Satz von der Hauptachsentransformation

Ist A eine reellwertige, symmetrische Matrix, dann gibt es eine orthogonale Matrix Q, sodass $QAQ^T = D$ eine Diagonalmatrix ist. Die Spalten von Q^T bilden eine Orthonormalbasis aus Eigenvektoren.

„Definition": Hauptachsentransformation

Wir nennen $A = Q^T D Q$ die Hauptachsentransformation von A, da die Eigenwerte von A die Diagonalelemente von D sind und zugehörige Eigenvektoren die Spalten von Q.

„Notation": Diagonalmatrix

Ist D eine Diagonalmatrix und y der Vektor der Diagonaleinträge, so schreiben wir auch $D = \operatorname{diag}(y)$.

„Bemerkung": Herkunft des Satznamens

Sind alle Eigenwerte der symmetrischen Matrix A positiv, so ist die Menge der Punkte $E := \{x \in \mathbb{R}^n \mid x^T A x \leq 1\}$ im $\mathbb{R}^n$ ein Ellipsoid.

Die orthogonale Transformation mit Q „dreht" dann die Achsen des Ellipsoids in die Koordinatenachsen.

Korollar 5.7.5

Die Spalten der Matrix Q bilden eine Orthonormalbasis von $\mathbb{R}^n$ bestehend aus Eigenvektoren von A.

Cholesky-Faktorisierung

Definition 5.8.1: positiv definit / positiv semidefinit

Sei $A \in \mathbb{R}^{n \times n}$ eine symmetrische Matrix. Wir nennen A positiv definit, falls $x^T A x > 0$ für alle $x \in \mathbb{R}^n \setminus \{0\}$.

Gilt nur $x^T A x \geq 0$ für alle $x \in \mathbb{R}^n$, so nennen wir A positiv semidefinit.

„Ziel"

Gleichungssystem $Ax = b$ für positiv definite Matrizen noch effizienter lösen durch Bestimmung einer „Wurzel" solcher Matrizen (untere Dreiecksmatrix L mit $A = LL^T$).

Proposition 5.8.2

Sei A eine reellwertige, symmetrische Matrix. Dann gilt:

A ist positiv definit $\iff$ der kleinste Eigenwert ist $\lambda_n > 0$.

„Definition": Hauptuntermatrix / führende Hauptuntermatrix

Sei A eine quadratische Matrix. Streichen wir einige Zeilen von A und die Spalten mit dem gleichen Index, so erhalten wir eine sogenannte Hauptuntermatrix der Form $A_{I,I}$ mit $I \subseteq \{1, \ldots, n\}$.

Streichen wir nur Zeilen und Spalten mit höchsten Indizes, so erhalten wir eine sogenannte führende Hauptuntermatrix.

Proposition 5.8.3

Sei $A \in \mathbb{R}^{n \times n}$ symmetrisch und positiv definit und $I \subseteq \{1, \ldots, n\}$. Dann ist auch $A_{I,I}$ symmetrisch und positiv definit.

Satz 5.8.4: Satz von der Cholesky-Faktorisierung

Sei $A \in \mathbb{R}^{n \times n}$ symmetrisch und positiv definit. Dann existiert eine eindeutig bestimmte, reguläre untere Dreiecksmatrix L mit $A = LL^T$ und $L_{ii} > 0$ für $i = 1, \ldots, n$.

„Ausführungen zum Cholesky-Algorithmus"

„Algorithmus aus Beweis zu Satz 5.8.4"

- löse jeweils die Gleichung $L_{I,I} c_I = b_I$ nach c auf
- ziehe dann aus $a_{i+1,i+1} - c_I^T c_I$ die Wurzel

„Betrachtung des Zustandekommens der einzelnen Einträge von A"

- gehe von $A = LL^T$ aus und betrachte für einzelne Rechenschritte der Cholesky-Faktorisierung das Zustandekommen der einzelnen Einträge von A.

- $a_{ij} = L_{i.} L_{j.}^T = \sum_{k=1}^{n} l_{ik} l_{jk} = \sum_{k=1}^{\min\{i,j\}} l_{ik} l_{jk}$.

- rekursiv Einträge von L ausrechnen, erhalten folgenden Algorithmus

$$l_{11} = \sqrt{a_{11}}, \quad l_{i1} = \frac{a_{i1}}{l_{11}}, \quad l_{22} = \sqrt{a_{22} - l_{21}^2},$$

$$l_{i2} = \frac{a_{i2} - l_{21} l_{i1}}{l_{22}}$$

- Allgemein: $l_{kk} = \sqrt{a_{kk} - \sum_{j=1}^{k-1} l_{kj}^2}$, $l_{ik} = \dfrac{a_{ik} - \sum_{j=1}^{k-1} l_{kj} l_{ij}}{l_{kk}}$ für $i = k+1, \ldots, n$

Algorithmus: Cholesky

```
def cholesky(A):
    n = A.shape[0]
    L = zeros((n,n))
    try:
      for k in range(n):
        L[k,k] = sqrt(A[k,k] - dot(L[k,:],L[k,:]))
        for l in range(k+1,n):       Skalarprodukt zweier Vektoren in NumPy
          L[l,k] = (A[l,k] - dot(L[l,:],L[k,:]))/L[k,k]
    except:
      print "Matrix nicht positiv definit"
    return L
```

„Aufgabe 5.8.6"

Sei A eine reguläre, symmetrische Matrix. Dann gilt:

a.) $\lambda \in \mathbb{R}$ ist Eigenwert von $A \iff \lambda^{-1}$ ist Eigenwert von A^{-1}

b.) A ist positiv definit $\iff A^{-1}$ ist positiv definit

„Aufgabe 5.8.7"

Sei $A \in \mathbb{R}^{n\times n}$ eine symmetrische Matrix. Dann gilt:

A ist nicht positiv definit $\iff$ im Verlauf des Cholesky-Verfahren ist für ein $1 \leq k \leq n$ der Ausdruck

$$a_{kk} - \sum_{i=1}^{k-1} l_{ki}^2 \leq 0.$$

Matrixnormen

Definition 5.9.1: Norm

Sei X ein Vektorraum über $\mathbb{R}$. Eine Abbildung $\|\cdot\|: X \to \mathbb{R}$ heißt Norm, wenn

(N1) $\|x\| = 0 \iff x = 0$

(N2) $\forall \alpha \in \mathbb{R}: \|\alpha x\| = |\alpha| \, \|x\|$ (Homogenität)

(N3) $\|x+y\| \leq \|x\| + \|y\|$ (Dreiecksungleichung)

Ein Vektorraum mit Norm heißt normierter Vektorraum.

Beispiel 5.9.2: euklidische Norm

Für $x \in \mathbb{R}^n$ ist die euklidische Norm $\|\cdot\|$ die aus dem Satz des Pythagoras bekannte Größe

$$\|x\|_2 := \sqrt{x^T x} = \sqrt{\sum_{i=1}^n x_i^2}.$$

Proposition 5.9.4

Seien $X = \mathbb{R}^n$ und $Y = \mathbb{R}^m$ normierte $\mathbb{R}$-Vektorräume der Dimensionen n bzw. m mit Normen $\|\cdot\|_X$, $\|\cdot\|_Y$.

Dann ist die Abbildung $\|\cdot\|_{X\to Y}: \mathbb{R}^{m\times n} \to \mathbb{R}$ mit

$$\|A\|_{X\to Y} := \sup_{x\in\mathbb{R}^n\setminus\{0\}} \frac{\|Ax\|_Y}{\|x\|_X} = \max_{\|x\|_X=1} \|Ax\|_Y$$

eine Norm auf dem Vektorraum der $(m\times n)$-Matrizen über $\mathbb{R}$, die natürliche Norm.

Beispiel 5.9.5

Wir betrachten die Normen $\|\cdot\|_1, \|\cdot\|_2, \|\cdot\|_\infty$. Sind dann $X = \mathbb{R}^n = Y$, so schreiben wir für die zugehörigen Matrixnormen $\|A\|_1, \|A\|_2, \|A\|_\infty$. Es gilt:

a.) $\|A\|_1 = \max_{j\in\{1,\dots,n\}} \sum_{i=1}^n |a_{ij}|$ (Spaltensummennorm)

b.) $\|A\|_\infty = \max_{i\in\{1,\dots,n\}} \sum_{j=1}^n |a_{ij}|$ (Zeilensummennorm)

c.) $\|A\|_2 = \max\{\sqrt{\lambda} \mid \lambda \text{ ist Eigenwert von } A^TA\}$ (Spektralnorm)

Kondition

„Motivation"

Wie wirken sich Datenfehler bei der Aufgabe $Ax=b$ mit regulärer Matrix A bei exakter Lösung auf den Vektor x aus?

„Ausführungen"

Sei Δb der absolute Fehler der rechten Seite.
Sei Δx der absolute Fehler des Lösungsvektors.

Somit fehlerbehaftete Gleichung $A(x+\Delta x) = b + \Delta b$.

Mit exakter Lösung x, also $Ax = b$ folgt

$\Delta x = A^{-1}\Delta b$ und somit $\|\Delta x\| \leq \|A^{-1}\| \, \|\Delta b\|$.

Mit $\|b\| \leq \|A\| \, \|x\|$ erhalten wir $\|x\| \geq \frac{\|b\|}{\|A\|}$.

Es folgt die folgende Abschätzung für den relativen Fehler:

$$\frac{\|\Delta x\|}{\|x\|} \leq \|A^{-1}\| \, \|A\| \frac{\|\Delta b\|}{\|b\|}.$$

Definition 5.10.1: Kondition

Sei $A \in \mathbb{R}^{n\times n}$ eine reguläre Matrix. Die Zahl $\mathrm{cond}(A) := \|A^{-1}\| \, \|A\|$ heißt Kondition der Matrix A.

Wir nennen A schlecht konditioniert, wenn die Konditionszahl $\mathrm{cond}(A)$ deutlich größer als 1 ist und gut konditioniert, wenn $\mathrm{cond}(A)$ nahe bei 1 ist.

Lemma 5.10.2

Sei $A \in \mathbb{R}^{n\times n}$ und $\|A\| < 1$. Dann ist $I_n + A$ regulär und

$$\frac{1}{1+\|A\|} \leq \|(I_n + A)^{-1}\| \leq \frac{1}{1-\|A\|}.$$

„Ausführungen"

Wir sind an den Auswirkungen einer Störung von A bei der Lösung von $Ax=b$ und damit an einer Abschätzung von $\mathrm{cond}(A+\Delta A)$ interessiert. Nennen wir $A+\Delta A$ dann B, so hilft uns folgendes Lemma weiter.

Lemma 5.10.3: Störungslemma

Seien $A, B \in \mathbb{R}^{n\times n}$ quadratisch, A regulär und $\|A^{-1}\| \, \|B-A\| < 1$. Dann ist auch B regulär und

$$\|B^{-1}\| \leq \frac{\|A^{-1}\|}{1 - \|A^{-1}\| \, \|B-A\|}.$$

Satz 5.10.4

Seien A, $\Delta A \in \mathbb{R}^{n\times n}$ und gelte $\|A^{-1}\| \, \|\Delta A\| < 1$.
Seien $x \in \mathbb{R}^n$, $x \neq 0$ bzw. $x + \Delta x \in \mathbb{R}^n$ Lösungen des Systems $Ax = b$ bzw. $(A+\Delta A)(x+\Delta x) = b$. Dann lässt sich der relative Fehler in x abschätzen durch den relativen Fehler in A und die Konditionszahl von A zu

$$\frac{\|\Delta x\|}{\|x\|} \leq \frac{\mathrm{cond}(A)}{1 - \mathrm{cond}(A)\frac{\|\Delta A\|}{\|A\|}} \, \frac{\|\Delta A\|}{\|A\|}.$$

Nichtlineare Optimierung

„Allgemeines Optimierungsproblem im $\mathbb{R}^n$"

$\min_{x \in S} f(x)$ mit $S \subseteq \mathbb{R}^n$ und reellwertiger Zielfunktion $f: S \to \mathbb{R}$

Mathematisch präziser müsste $\inf_{x \in S} f(x)$ statt $\min_{x \in S} f(x)$ schreiben.

„Bemerkung"

Wegen $\max_{x \in S} f(x) = -\min_{x \in S} (-f)(x)$ können wir uns auf die Betrachtung von Minimierungsproblemen beschränken.

Bemerkung 6.0.1

Bei den folgenden Betrachtungen verlangen wir stetige Funktionen, die sogar ein- oder zweimal stetig differenzierbar sind. Bei den betrachteten Zulässigkeitsbereichen sollte man sich auf zusammenhängende und offene / abgeschlossene Mengen beschränken.

Bemerkung 6.0.2

Es wird ohne Nachweis benutzt, dass stetige Funktionen auf beschränkten abgeschlossenen Mengen ihre Extremwerte stets annehmen.

Lokale Minima bei Funktionen einer Variablen

„Definition": Landau-Symbol o

Ableitungen können mittels des Landau-Symbols o charakterisiert werden. Für zwei reellwertige Funktionen $f: \mathbb{R} \to \mathbb{R}$ und $g: \mathbb{R} \to \mathbb{R}$ schreiben wir $f = o(g)$, falls für jede Nullfolge t_n gilt $f(t_n) = o(g(t_n))$, also falls $\lim_{h \to 0} \frac{f(h)}{g(h)} = 0$.

Proposition 6.1.1: zwei Spezialfälle des Satzes von Taylor

Sei $f: S \to \mathbb{R}$ eine zweimal stetig differenzierbare Funktion, wobei $S \subseteq \mathbb{R}$ offen sei. Dann gilt:

a.) $f'(x) = r \iff f(x+h) = f(x) + rh + o(h)$

b.) $f''(x) = q \iff f(x+h) = f(x) + f'(x)h + \frac{1}{2}qh^2 + o(h^2)$

Definition 6.1.2: (strikter) lokales Minimum

Seien $S \subseteq \mathbb{R}$, $f: S \to \mathbb{R}$ und $x^* \in S$. Wir sagen, f hat an der Stelle x^* ein lokales (oder relatives) Minimum über S, wenn gilt
$\exists \varepsilon > 0 \; \forall x \in U_\varepsilon(x^*): f(x) \geq f(x^*)$.

Wir nennen die Stelle x^* ein striktes lokales Minimum, wenn gilt
$\exists \varepsilon > 0 \; \forall x \in U_\varepsilon(x^*)$ mit $x \neq x^*$: $f(x) > f(x^*)$.

Definition: (strikter) globales Minimum

Seien $S \subseteq \mathbb{R}$, $f: S \to \mathbb{R}$ und $x^* \in S$. Wir sagen, f hat an der Stelle x^* ein globales Minimum, wenn gilt
$\forall x \in S: f(x) \geq f(x^*)$.

Wir nennen die Stelle x^* ein striktes globales Minimum, wenn gilt
$\forall x \in S$ mit $x \neq x^*$: $f(x) > f(x^*)$.

Satz 6.1.3: notwendige Kriterien bzw. hinreichendes Kriterium für lokale Minima

Sei $S \subseteq \mathbb{R}$ eine offene Menge und $f: S \to \mathbb{R}$ eine zweimal stetig differenzierbare Funktion.

a.) Hat f an der Stelle x^* ein lokales Minimum, so ist $f'(x^*) = 0$.

b.) Hat f an der Stelle x^* ein lokales Minimum, so ist $f'(x^*) = 0$ und $f''(x^*) \geq 0$.

c.) Gilt $f'(x^*) = 0$ und $f''(x^*) > 0$, so hat f an der Stelle x^* ein striktes lokales Minimum.

Abbildungen in mehreren Veränderlichen

Definition 6.2.1: Skalarfunktion / Vektorfeld

Sei $S \subseteq \mathbb{R}^n$ und sei $f: S \to \mathbb{R}^m$ eine Abbildung.

Ist $m = 1$, so nennen wir f eine Skalarfunktion und schreiben sie als $f(x) = f(x_1, \dots, x_n)$.

Ist $m \geq 2$, so nennen wir f ein Vektorfeld. Jede Koordinate im Bild ist dann eine Skalarfunktion und wir können f schreiben als

$$f(x) = \begin{pmatrix} f_1(x) \\ \vdots \\ f_m(x) \end{pmatrix} = \begin{pmatrix} f_1(x_1, \dots, x_n) \\ \vdots \\ f_m(x_1, \dots, x_n) \end{pmatrix}.$$

„Bemerkung"

Für die Verallgemeinerung von Stetigkeit und Differenzierbarkeit für mehrdimensionale Funktionen benötigen wir Normen als Ersatz für den Betrag. Für das konkrete Rechnen verwenden wir i.d.R. die euklidische Norm.

Definition 6.2.3: Konvergenz

Sei $(x_i)_{i \in \mathbb{N}}$ eine Folge, wobei für alle $i \in \mathbb{N}$ $x_i \in \mathbb{R}^n$ gelte und sei $x^* \in \mathbb{R}^n$. Wir sagen die Folge konvergiert gegen x^*, wenn $\lim_{i \to \infty} \|x^* - x_i\| = 0$.

Definition 6.2.3: ε-Umgebung

Sei $x^* \in \mathbb{R}^n$ und sei $\varepsilon > 0$. Dann ist die ε-Umgebung von x^* definiert als
$U_\varepsilon(x^*) := \{x \in \mathbb{R}^n \mid \|x^* - x\| < \varepsilon\}$.

Definition 6.2.3: offene und abgeschlossene Mengen

Die Menge $U \subseteq \mathbb{R}^n$ heißt offen, wenn zu jedem $x \in U$ ein $\varepsilon > 0$ mit $U_\varepsilon(x) \subseteq U$ existiert.

Die Menge $A \subseteq \mathbb{R}^n$ heißt abgeschlossen, wenn $\mathbb{R}^n \setminus A$ offen ist.

Definition 6.2.5: Stetigkeit

Sei $U \subseteq \mathbb{R}^n$ offen und $f: U \to \mathbb{R}^n$ eine Abbildung und sei $x^* \in U$. Wir sagen f ist stetig in x^*, wenn
$\forall \varepsilon > 0 \; \exists \delta > 0 \; \forall x \in U_\delta(x^*) \cap U: f(x) \in U_\varepsilon(f(x^*))$.

Ist f stetig in allen $x \in U$, so nennen wir f stetig.

„Aufgabe 6.2.7": Stetigkeit

Sei U eine offene Menge, $f: U \to \mathbb{R}^n$ eine Funktion und $x^* \in U$. Dann gilt:

f ist stetig in $x^* \iff$ Für jede Folge $(x_i)_{i \in \mathbb{N}}$ in U gilt: $\lim_{i \to \infty} x_i = x^* \Rightarrow \lim_{i \to \infty} f(x_i) = f(x^*)$.

Definition 6.2.8: lokales Minimum für Funktionen mehrerer Variablen

Seien $S \subseteq \mathbb{R}^n$, $f: S \to \mathbb{R}$ und $x^* \in S$. Dann sagen wir, f hat an der Stelle x^* ein lokales Minimum über S, wenn es ein $\varepsilon > 0$ gibt mit
$\forall x \in S \cap U_\varepsilon(x^*): f(x) \geq f(x^*)$.

Gilt sogar
$\forall x \in S \cap U_\varepsilon(x^*)$ mit $x \neq x^*$: $f(x) > f(x^*)$,
so liegt an der Stelle x^* ein striktes lokales Minimum vor.

Definition 6.2.8: globales Minimum für Funktionen mehrerer Variablen

Seien $S \subseteq \mathbb{R}^n$, $f: S \to \mathbb{R}$ und $x^* \in S$.

Gilt $\forall x \in S: f(x) \geq f(x^*)$, so hat f an der Stelle x^* ein globales Minimum.

Steilkurs mehrdimensionale Differentialrechnung

Partielle und totale Ableitungen

„Ziel / Motivation"

Konstruktion eines mehrdimensionalen Ableitungsbegriffs, um die Kriterien für die Existenz von lokalen Extremwerten aus dem Eindimensionalen auf Funktionen mehrerer Veränderlicher zu übertragen.

Definition 6.3.1: i-te partielle Ableitung von f

Sei $S \subseteq \mathbb{R}^n$ offen, $x^* = (x_1^*, \dots, x_n^*) \in S$ und $1 \leq i \leq n$.
Wir betrachten nun die Funktion $g(t): I \to \mathbb{R}$ definiert durch

$$g(t) = f(x_1^*, \dots, x_{i-1}^*, x_i^* + t, x_{i+1}^*, \dots, x_n^*),$$

wobei I ein offenes Intervall mit $0 \in I$ und $\{x^* + te_i \mid t \in I\} \subseteq S$ sei. Ist g dann an der Stelle 0 differenzierbar, so nennen wir

$$\frac{\partial f}{\partial x_i}(x^*) := g'(0)$$

die i-te partielle Ableitung von f.

Der Gradient $\nabla f(x^*)$ von f in x^* ist der Zeilenvektor (!) der partiellen Ableitungen

$$\left(\frac{\partial f}{\partial x_1}(x^*), \dots, \frac{\partial f}{\partial x_n}(x^*)\right).$$

Der Gradient $\nabla f: S \to \mathbb{R}^n$ ist also eine vektorwertige Abbildung, wobei wir hier aber Zeilenvektoren im Bild haben.

„Bemerkung"

Zur Bildung partieller Ableitungen betrachten wir also die anderen Variablen einfach als Konstanten und gehen so vor, wie aus dem Umgang mit Funktionen einer Variablen gewohnt.

„Bemerkung"

Im Allgemeinen versuchen wir mit dem Differenzieren eine Abbildung lokal möglichst gut durch eine (affin) lineare Abbildung zu approximieren. Lineare Abbildungen $L: \mathbb{R}^n \to \mathbb{R}^l$ werden (bzgl. der Einheitsvektoren als Standardbasis) durch (l,n)-Matrizen beschrieben. Also erwarten wir als Abbildung einer Funktion $h: \mathbb{R}^n \to \mathbb{R}^l$ eine Matrix dieser Größe.

Definition 6.3.3: Jacobische Jh

Ist $h: U \to \mathbb{R}^l$ eine vektorwertige Abbildung, so bezeichnen wir mit Jh die Jacobische, also die Matrix der partiellen Ableitungen

$$Jh(x) := \begin{pmatrix} \frac{\partial h_1}{\partial x_1} & \dots & \frac{\partial h_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial h_l}{\partial x_1} & \dots & \frac{\partial h_l}{\partial x_n} \end{pmatrix}.$$

„Definition": stetig differenzierbar

Wir nennen eine reellwertige Funktion f stetig differenzierbar, wenn alle partiellen Ableitungen existieren.

„Definition": zweite partielle Ableitung

$$\frac{\partial^2 f}{\partial x_j \partial x_i} := \frac{\partial\left(\frac{\partial f}{\partial x_i}\right)}{\partial x_j} =: \frac{\partial}{\partial x_j}\left(\frac{\partial f}{\partial x_i}\right).$$

Für $\frac{\partial^2 f}{\partial x_j \partial x_i}$ schreiben wir auch kürzer $\frac{\partial^2 f}{\partial x_i^2}$.

„Definition": zweimal stetig differenzierbar

Wir nennen eine reellwertige Funktion f zweimal stetig differenzierbar, wenn alle zweiten partiellen Ableitungen existieren und stetig sind.

„Definition": Hessematrix $\nabla^2 f(x)$

Sei f eine reellwertige Funktion. Wir nennen die Jacobische des Gradienten Hessematrix $\nabla^2 f(x)$.
Die Hessematrix ist also die Matrix der zweiten partiellen Ableitungen

$$\nabla^2 f(x) := \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \dots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \dots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix}.$$

Existieren alle zweiten partiellen Ableitungen einer Funktion $f: S \to \mathbb{R}$ und sind stetig, so ist die Hessematrix $\nabla^2 f(x)$ eine symmetrische Matrix, da der Satz von Schwarz gilt.

Satz 6.3.5: Satz von Schwarz

Ist $S \subseteq \mathbb{R}^2$ offen und $f: S \to \mathbb{R}$ stetig differenzierbar. Existiert dann die zweite partielle Ableitung $\frac{\partial^2 f}{\partial x \partial y}$ und ist stetig, so existiert auch die partielle Abbildung $\frac{\partial^2 f}{\partial y \partial x}$ und es gilt

$$\frac{\partial^2 f}{\partial y \partial x} = \frac{\partial^2 f}{\partial x \partial y}.$$

Satz 6.3.7: mehrdimensionaler Satz von Taylor

Sei $S \subseteq \mathbb{R}^n$ offen und seien $f, g: S \to \mathbb{R}$ Funktionen, dabei seien f, g stetig differenzierbar und bei g sei darüber hinaus der Gradient stetig differenzierbar. Ist dann $x \in S$ und $v \in \mathbb{R}^n$ mit $x + v \in S$, so gilt

a.) $f(x+v) = f(x) + \nabla f(x) \cdot v + o(\|v\|)$

b.) $g(x+v) = g(x) + \nabla g(x) \cdot v + \frac{1}{2} v^T \nabla^2 g(x) v + o(\|v\|)^2$

Proposition 6.3.8: mehrdimensionale Summen- und Produktregel

Sei $S \subseteq \mathbb{R}^n$ offen und seien $f_1, f_2: S \to \mathbb{R}^m$ und $g_1, g_2: S \to \mathbb{R}$ stetig differenzierbare Funktionen und $x \in S$.
Dann gilt:

a) $J(f_1 + f_2)(x) = Jf_1(x) + Jf_2(x)$

b.) $\nabla(g_1 g_2)(x) = g_2(x)\nabla g_1(x) + g_1(x)\nabla g_2(x)$.

Satz 6.3.9: mehrdimensionale Kettenregel

Seien $S \subseteq \mathbb{R}^n$ und $g: S \to \mathbb{R}^m$ stetig differenzierbar in $x^* \in S$ sowie $U \subseteq \mathbb{R}^m$ und $f: U \to \mathbb{R}^k$ differenzierbar in $g(x^*) \in U$.
Dann ist auch $f \circ g$ differenzierbar in x^* und es gilt

$$J(f \circ g)(x^*) = Jf(g(x^*))\, Jg(x^*).$$

Kurven

„Ziel / Motivation"

Mit partiellen Ableitungen haben wir das Verhalten von Abbildungen $f: \mathbb{R}^n \to \mathbb{R}$ in Richtung der Koordinatenachsen gut im Griff. Falls Nebenbedingungen uns daran hindern in diese Richtung zu laufen, müssen wir nur auf gekrümmten Kurven laufen dürfen. Durch das Einführen von Wegen $c: I \to \mathbb{R}^n$ können wir solche Funktionen $f: \mathbb{R}^n \to \mathbb{R}$ auf Funktionen einer Veränderlichen zurückführen. Die Komposition $f \circ c: I \to \mathbb{R}$ ist nämlich eine reelle Funktion in einer Variablen.

Definition 6.3.11.a: Kurve, Weg und Komponentenfunktionen

Sei $I \subseteq \mathbb{R}$ ein Intervall. Eine Kurve oder ein Weg im $\mathbb{R}^n$ ist eine stetige Abbildung $c: I \to \mathbb{R}^n$ mit $t \mapsto c(t) = (c_1(t), \dots, c_n(t))^T$.
Insbesondere sind also alle Komponentenfunktionen $c_i: I \to \mathbb{R}$ stetige, reellwertige Funktionen.

Definition 6.3.11.b: k-fach stetig differenzierbar / Tangentialvektor

Wir nennen eine Kurve k-fach stetig differenzierbar, wenn alle Komponentenfunktionen k-fach stetig differenzierbar sind für $k \in \mathbb{N}$.

Ist $t_0 \in I$, so nennen wir $c'(t_0) := (c_1'(t_0), \dots, c_n'(t_0))^T$ den Tangentialvektor an c in t_0, wobei $c_i'(t_0)$ die Ableitungen der Komponentenfunktionen in t_0 sind.

Proposition 6.3.14

Sei $S \subseteq \mathbb{R}^n$, $f: S \to \mathbb{R}$ und $x^* \in S$. Ist x^* ein lokales Minimum (striktes lokales Minimum) von f, so gilt für jedes $\eta > 0$ und jeden stetig differenzierbaren Weg $c: [0, \eta] \to S$ mit $c(0) = x^*$ und $c'(0) \neq 0$:

$$\exists 0 < \delta \leq \eta \; \forall 0 < t \leq \delta: (f \circ c)(t) := f(c(t)) \geq f(x^*) \text{ bzw. } f(c(t)) > f(x^*).$$

Mit Worten: Es gibt einen Zeitpunkt δ mit $0 < \delta \leq \eta$, bis zu dem man entlang c zu keinem tieferen Punkt als x^* gelangt.

„Die lokalen Minima sind die „Täler" in der Landschaft. Im Tal, so geht es in alle Richtungen bergauf oder zumindest in keine Richtung bergab."

Definition 6.3.16: Richtungsableitung

Sei $c: I \to \mathbb{R}^n$ eine stetig differenzierbare Kurve und $f: S \to \mathbb{R}$ mit $p = c(t_0) \in S$ und $c'(t_0) = d \neq 0$. Existiert dann $(f \circ c)'(t_0)$, so heißt diese Zahl die Richtungsableitung $\frac{\partial f}{\partial d}$ von f in Richtung d im Punkt p.

Proposition 6.3.17: Richtungsableitung ist unabhängig von dem speziellen Weg

Ist f stetig differenzierbar, so ist $\frac{\partial f}{\partial d}(p) = \nabla f(p) \cdot d$.

Satz 6.3.18

Sei $p \in S \subseteq \mathbb{R}^n$ und $f: S \to \mathbb{R}$ zweimal stetig differenzierbar. Sei c ein zweimal stetig differenzierbarer Weg mit $c(t_0) = p$ und $c'(t_0) = d \neq 0$. Dann ist

$$(f \circ c)(t) = f(p) + \nabla f(p) d (t - t_0) + \frac{1}{2} \nabla f(p) c''(t_0)(t - t_0)^2 + \frac{1}{2} d^T \nabla^2 f(p) d (t - t_0)^2 + o((t - t_0)^2).$$

Notwendige und hinreichende Bedingungen für Extremwerte

Definition 6.4.1.a: Notation $C^k(M)$

Ist $M \subseteq \mathbb{R}^n$ offen und f k-fach stetig differenzierbar in M, d.h. alle k-fachen partiellen Ableitungen existieren und sind stetig, so schreiben wir dafür $f \in C^k(M)$.

Definition 6.4.1.b: zulässige Richtung

Seien $S \subseteq \mathbb{R}^n$, $x \in S$ und $d \in \mathbb{R}^n$. Dann heißt d zulässige Richtung für x bzgl. S, wenn es ein $\varepsilon > 0$ und einen stetig differenzierbaren Weg $c: [0, \varepsilon] \to S$ gibt mit $c(0) = x$ und $c'(0) = d$.

Definition 6.4.1.c: zulässige Abstiegsrichtung

Seien $S \subseteq M \subseteq \mathbb{R}^n$, M offen, $x \in S$ und $d \in \mathbb{R}^n$. Sei zusätzlich $f \in C^1(M)$. Wir nennen d zulässige Abstiegsrichtung in x bzgl. S, wenn d zulässige Richtung für x bzgl. S ist und darüber hinaus $\nabla f(x) d < 0$ ist.

Proposition 6.4.2: Notwendige Bedingung erster Ordnung für lokale Minima

Seien $S \subseteq M \subseteq \mathbb{R}^n$, M offen und $f \in C^1(M)$. Ist x^* ein relatives Minimum von f in S, so gilt für jede zulässige Richtung d für x^* bzgl. S:

$$\nabla f(x^*) d \geq 0.$$

Korollar 6.4.3

Ist $S \subseteq \mathbb{R}^n$ offen und $x^* \in S$ ein relatives Maximum von f, so ist $\nabla f(x^*) = 0$.

Proposition 6.4.5: Notwendige Bedingungen zweiter Ordnung für lokale Minima

Sei $M \subseteq \mathbb{R}^n$ offen und $S \subseteq M$ sowie $f \in C^2(M)$. Ist dann x^* ein relatives Minimum von f in S, so gilt für jedes d, für das es ein ε gibt mit $x^* + \alpha d \in S$ für $0 \leq \alpha \leq \varepsilon$:

a.) $\nabla f(x^*) d \geq 0$

b.) $\nabla f(x^*) d = 0 \Rightarrow d^T \nabla^2 f(x^*) d \geq 0$.

Korollar 6.4.6

Ist x^* ein relatives Minimum von f im Innern von S, so gilt für alle $d \in \mathbb{R}^n$:

a.) $\nabla f(x^*) = 0$

b.) $d^T \nabla^2 f(x^*) d \geq 0$. „Hessematrix ist positiv semidefinit (Definition 5.5.1)"

Proposition 6.4.7: Hinreichende Bedingungen zweiter Ordnung

Sei $M \subseteq \mathbb{R}^n$ offen, $S \subseteq M$, $f \in C^2(M)$ und $x^* \in S$. Gilt dann

a.) $\nabla f(x^*) = 0$ und

b.) $\nabla^2 f(x^*)$ ist positiv definit,

so ist x^* ein striktes lokales Minimum von f.

Satz 6.4.8

Sei $x^* \in M \subseteq \mathbb{R}^n$ offen und $f \in C^1(M)$ eine Funktion. Gibt es dann für alle $d \in \mathbb{R}^n \setminus \{0\}$ ein $\alpha_d > 0$, so dass 0 striktes lokales Minimum der Funktion $f_d: [0, \alpha_d] \to \mathbb{R}$, definiert durch $f_d(t) := f(x^* + td)$, ist, so ist x^* striktes lokales Minimum von f.

Exkurs Mannigfaltigkeiten und Tangentialräume

Siehe S. 258 ff.

Definition 6.5.1: Tangentialraum

Sei M eine Mannigfaltigkeit (also Lösungsmenge eines Systems von Gleichungen $h(x) = 0$) und $x^* \in M$. Dann bezeichnen wir die Menge

$$T_{x^*}^M := \{ c'(0) \mid \exists c: I \to M, \text{ stetig differenzierbar mit } 0 \in I \text{ und } c(0) = x^* \}$$

als Tangentialraum an M in x^*.

„Bemerkung": orthogonales Komplement

Der Tangentialraum an eine Mannigfaltigkeit in einem Punkt x^*, die durch eine einzelne Gleichung definiert wird, steht senkrecht auf dem Gradienten der definierenden Gleichung, er bildet das orthogonale Komplement.

Bedingungen für Extrema auf gleichungsdefinierten Mengen

Satz 6.6.1

Seien $h: \mathbb{R}^n \to \mathbb{R}^k$, $f: \mathbb{R}^n \to \mathbb{R}$ zweimal stetig differenzierbare Funktionen, $k \leq n$ und $x^* \in S$ sei ein lokales Minimum der Optimierungsaufgabe $\min f(x)$ unter $h(x) = 0$. Ferner seien $\nabla h_1(x^*), \dots, \nabla h_k(x^*)$ linear unabhängig.

Dann gibt es $\lambda_1, \dots, \lambda_k \in \mathbb{R}$ mit

$$\nabla f(x^*) = \lambda_1 \nabla h_1(x^*) + \dots + \lambda_k \nabla h_k(x^*).$$

Die λ_i nennt man Lagrange'sche Multiplikatoren.

Definition 6.6.2: positiv semidefinit

Sei $Q \in \mathbb{R}^{n \times n}$ eine symmetrische Matrix und V ein Untervektorraum von $\mathbb{R}^n$. Dann nennen wir Q positiv semidefinit auf V, wenn $\forall v \in V: v^T Q v \geq 0$ gilt.

Analog übertragen wir die Begriffe positiv definit, negativ semidefinit, negativ definit und indefinit auf V.

„Definition": Kern

Sei $A \in \mathbb{R}^{m \times n}$ eine Matrix. Wir bezeichnen den Kern von A mit $\ker(A) := \{ x \in \mathbb{R}^n \mid Ax = 0 \}$.

Satz 6.6.4: Notwendige Bedingungen zweiter Ordnung für lokale Minima

Unter den Voraussetzungen von Satz 6.6.1 gilt:

Ist $x^* \in S$ ein lokales Minimum der Optimierungsaufgabe, so gibt es $\lambda_1, \dots, \lambda_k \in \mathbb{R}$ mit

a.) $\nabla f(x^*) = \sum_{i=1}^{k} \lambda_i \nabla h_i(x^*)$ und

b.) die Matrix $L := \nabla^2 f(x^*) - \sum_{i=1}^{k} \lambda_i \nabla^2 h_i(x^*)$ ist positiv semidefinit auf dem Tangentialraum von $S = \{ x \in \mathbb{R}^n \mid h(x) = 0 \}$ in x^*. Das heißt, dass für alle $d \in \ker(\nabla h(x^*))$ gilt:

$$d^T L d \geq 0.$$

Satz 6.6.5: Hinreichende Bedingungen zweiter Ordnung für lokale Minima

Seien $h: \mathbb{R}^n \to \mathbb{R}^k$, $f: \mathbb{R}^n \to \mathbb{R}$ zweimal stetig differenzierbare Funktionen und $k \leq n$. Gelte $h(x^*) = 0$, $\nabla f(x^*) = \sum_{i=1}^{k} \lambda_i \nabla h_i(x^*)$ für einen Vektor λ.
Sei ferner die Matrix $L := \nabla^2 f(x^*) - \sum_{i=1}^{k} \lambda_i \nabla^2 h_i(x^*)$ positiv definit auf dem Tangentialraum von $S := \{x \in \mathbb{R}^n \mid h(x) = 0\}$ in x^*. Dann ist x^* strikte lokales Minimum der Optimierungsaufgabe min $f(x)$ unter $h(x) = 0$.

Bedingungen für Extrema auf ungleichungsdefinierten Mengen

Satz 6.7.1: Karush-Kuhn-Tucker-Bedingungen

Seien $f: \mathbb{R}^n \to \mathbb{R}$, $h: \mathbb{R}^n \to \mathbb{R}^k$ und $g: \mathbb{R}^n \to \mathbb{R}^l$ stetig differenzierbar. Sei x^* ein regulärer Punkt der Nebenbedingungen (Definition siehe unten) und ein relatives Minimum des Problems

$$\min f(x) \text{ unter } h(x) = 0,\ g(x) \leq 0.$$

Dann gibt es Koeffizienten $\lambda_1, \dots, \lambda_k \in \mathbb{R}$ und $\mu_1, \dots, \mu_l \in \mathbb{R}$ mit $\mu_j \leq 0$ mit

$$\nabla f(x^*) = \sum_{i=1}^{k} \lambda_i \nabla h_i(x^*) + \sum_{j=1}^{l} \mu_j \nabla g_j(x^*) \quad \text{und}$$

$$\sum_{j=1}^{l} \mu_j g_j(x^*) = 0.$$

Definition 6.7.2: aktiv

In der Situation von Satz 6.7.1 heißt eine Ungleichungsnebenbedingung $g_j(x) \leq 0$ aktiv in x^*, wenn $g_j(x^*) = 0$ ist.

Satz 6.7.4

Sind die Voraussetzungen von Satz 6.7.1 erfüllt, so gilt außerdem:
Die Matrix $\nabla^2 f(x^*) - \sum_{i=1}^{k} \lambda_i \nabla^2 h_i(x^*) - \sum_{j=1}^{l} \mu_j \nabla^2 g_j(x^*)$ ist positiv semidefinit auf dem Tangentialraum der aktiven Nebenbedingungen von x^*.

Satz 6.7.6

Seien $f: \mathbb{R}^n \to \mathbb{R}$, $h: \mathbb{R}^n \to \mathbb{R}^k$ und $g: \mathbb{R}^n \to \mathbb{R}^l$ stetig differenzierbar und x^* ein regulärer Punkt. Ferner gebe es Koeffizienten $\lambda_1, \dots, \lambda_k \in \mathbb{R}$ und $\mu_1, \dots, \mu_l \in \mathbb{R}$, $\mu_i \leq 0$ mit

$$\nabla f(x^*) = \sum_{i=1}^{k} \lambda_i \nabla h_i(x^*) + \sum_{j=1}^{l} \mu_j \nabla g_j(x^*) \quad \text{und} \qquad (6.5)$$

$$\sum_{j=1}^{l} \mu_j g_j(x^*) = 0, \qquad (6.6)$$

sodass die Matrix

$$L := \nabla^2 f(x^*) - \sum_{i=1}^{k} \lambda_i \nabla^2 h_i(x^*) - \sum_{j=1}^{l} \mu_j \nabla^2 g_j(x^*)$$

positiv definit auf dem Tangentialraum der aktiven Nebenbedingungen, die in Gleichung (6.5) mit negativem Koeffizienten auftauchen, also auf dem Raum

$$M' := \{y \in \mathbb{R}^n \mid \nabla h_i(x^*) y = 0 \text{ für alle } i \text{ und } \nabla g_j(x^*) y = 0, \text{ falls } \mu_j < 0\}, \text{ ist.}$$

Dann ist x^* ein striktes lokales Minimum des Optimierungsproblems min $f(x)$ unter $h(x) = 0$, $g(x) \leq 0$.

Numerische Verfahren zur Nichtlinearen Optimierung

„Allgemein"

- Zentrale Stichworte: Suchrichtung und Schrittweite
- Wesentliches Vorgehen vieler Algorithmen der Nichtlinearen Optimierung:
  ausgehend von einem Iterationspunkt x_k
  ↳ bestimme eine Suchrichtung d_k
  ↳ bestimme Schrittweite λ_k
  ↳ setze $x_{k+1} = x_k + \lambda_k d_k$
- Strategie: in Abstiegsrichtung bewegen → Frage Schrittweitensteuerung
- Suchproblem bei festgelegter Suchrichtung d_k:
  Optimierung von $\tilde{f}: \mathbb{R}_+ \to \mathbb{R}$ mit $\tilde{f}(\alpha) = f(p_0 + \alpha d_k)$

Allgemeines Suchverfahren

„Allgemein"

- Keine allgemeine Aussage über globale Extremwerte bei allgemeinen Funktionen $f: \mathbb{R} \to \mathbb{R}$, die uns nur durch eine Unterroutine gegeben sind.
- „gute" Suchverfahren sollte mit möglichst wenigen Funktionsauswertungen auskommen
- Zur Klassifizierung beschränken wir uns zunächst auf gutartige Funktionen (hier: stetige Funktionen mit eindeutigem, globalen Minimum)

Definition 7.1.1: strikt unimodal

Sei $[a,b] \subseteq \mathbb{R}$ ein Intervall und $f: [a,b] \to \mathbb{R}$ eine Funktion. Dann heißt f strikt unimodal auf $[a,b]$, wenn f genau ein lokales Minimum in $[a,b]$ hat.

Proposition 7.1.3

Sei $[a,b] \subseteq \mathbb{R}$ ein Intervall und $f: [a,b] \to \mathbb{R}$ eine stetige Funktion. Dann gilt:

f ist strikt unimodal $\iff \forall a \leq x < y \leq b$ und $\lambda \in]0,1[$:
$f(\lambda x + (1-\lambda) y) < \max\{f(x), f(y)\}$.

„Definition ~~Beispiele~~ 7.1.4": strikt konvex

Wir nennen eine Funktion $f: \mathbb{R} \to \mathbb{R}$ strikt konvex, wenn für $s < t$ und $\lambda \in]0,1[$ gilt:
$f(\lambda s + (1-\lambda) t) < \lambda f(s) + (1-\lambda) f(t)$.

Definition 7.1.5.a: abgeschlossene Verbindungsstrecke

Sei $S \subseteq \mathbb{R}^n$ und seien $x, y \in S$. Die abgeschlossene Verbindungsstrecke $[x,y]$ zwischen x und y ist dann definiert als
$[x,y] = \{\lambda x + (1-\lambda) y \mid \lambda \in [0,1]\}$.

Definition 7.1.5.b: offene Verbindungsstrecke

Sei $S \subseteq \mathbb{R}^n$ und seien $x, y \in S$. Die offene Verbindungsstrecke $]x,y[$ zwischen x und y definieren wir als
$]x,y[= \{\lambda x + (1-\lambda) y \mid \lambda \in]0,1[\}$.

Definition 7.1.5.c: (strikt) konvexe Funktion

Sei $S \subseteq \mathbb{R}^n$. Eine Funktion $f: S \to \mathbb{R}$ heißt

a.) konvex, wenn S
  S ist konvex, also $\forall x,y \in S: [x,y] \subseteq S$ und
  $\forall x,y \in S \ \forall \lambda \in]0,1[: f(\lambda x + (1-\lambda) y) \leq \lambda f(x) + (1-\lambda) f(y)$.

b.) strikt konvex, wenn die letzte Ungleichung für $x \neq y$ stets strikt ist.

Bemerkung 7.1.6

Für $x = y$ ist $]x,y[= \{x\}$.

„Bemerkung": konvexe Funktionen

Schulwissen: konvexe Funktionen sind Funktionen, deren zweite Ableitung positiv ist und bei denen der Funktionsgraph stets oberhalb aller Tangenten verläuft.

„Aufgabe 7.1.7.a": Epigraph

Sei $S \subseteq \mathbb{R}^n$ eine konvexe Menge und sei $f: S \to \mathbb{R}$. Wir bezeichnen die Menge

$$\operatorname{epi}(f) := \left\{ \begin{pmatrix} x \\ \xi \end{pmatrix} \in \mathbb{R}^{n+1} \;\middle|\; x \in S, \ \xi \geq f(x) \right\}$$

als Epigraph der Funktion f. Es gilt:

$f: S \to \mathbb{R}$ ist konvex $\iff \operatorname{epi}(f)$ ist eine konvexe Menge.

„Aufgabe 7.1.7.b"

Sei $S \subseteq \mathbb{R}^n$ eine konvexe Menge und sei $f: S \to \mathbb{R}$ eine stetig differenzierbare Funktion. Dann gilt:
f ist konvex $\iff \forall x,y \in S: f(y) \geq f(x) + \nabla f(x)(y-x)$.

Proposition 7.1.8

Sei $f: [a,b] \to \mathbb{R}$ strikt unimodal und stetig und $a < x < y < b$. Dann gilt:

a.) $f(x) \geq f(y) \implies \min_{[a,b]} f = \min_{[x,b]} f$

b.) $f(x) \leq f(y) \implies \min_{[a,b]} f = \min_{[a,y]} f$.

„Merkregel"

Minimieren konvexer Funktionen über konvexen Mengen oder strikt unimodaler Funktionen ist eine gutartige Aufgabenstellung.

„Beschreibung allgemeines Suchverfahren"

- Wähle zwei Testpunkte x, y mit $a < x < y < b$ für $[a,b]$
- Fall 1: $f(x) \geq f(y)$
  ↳ verkleinere Suchintervall zu $[x,b]$
  Fall 2: $f(x) < f(y)$
  ↳ verkleinere Suchintervall zu $[a,y]$
- Abbruch, wenn Suchintervall relativ zur Zahlengröße oder Intervalllänge hinreichend klein geworden ist.

Algorithmus 7.1.10: Das allgemeine Suchverfahren

Sei $f: [a,b] \to \mathbb{R}$ eine strikt unimodale Funktion und $x \in [a,b]$. (erster Auswertungspunkt)

```
def findmin(f, a, x, b):
  laenge = abs(a) + abs(b)
  fx = f(x)
  while (b-a)/laenge >= eps:
    x, y, fx, fy = choosepoint(f, a, x, b, fx)
    if fx >= fy:          Wahl von Punkt y
      a = x               Verhältnis von x und y
      x = y
      fx = fy
    else:
      b = y
  return (a+b)/2
```

Spezielle Suchverfahren

Siehe Ausführungen zur Binärsuche auf Seite 294 f.

Proposition 7.2.1

Sei S ein Suchverfahren und für $k \in \mathbb{Z}_+$

$$S_k := \max\left\{ \frac{b_k - a_k}{b - a} \;\middle|\; \begin{array}{l} a_k, b_k \text{ nach } k\text{-ter Iteration bei} \\ \text{strikt unimodalen } f \end{array} \right\}.$$

Dann gilt $S_k \leq S_{k+1} + S_{k+2}$.

„Bemerkung"

Im günstigsten Fall erfüllen alle Intervalllängen die Rekursion $S_k = S_{k+1} + S_{k+2}$.

Im letzten Iterationsschritt sollten die Intervalle am besten gleich groß und $< \varepsilon$ sein.

Definition 7.2.2 : Fibonaccizahlen

Sei die Folge F_i definiert durch die Rekursionsgleichung $F_{i+2} = F_{i+1} + F_i$ für $i \in N$ und die Anfangsbedingungen $F_0 = F_1 = 1$. Dann heißen die F_i Fibonaccizahlen.

„Aufgabe 7.2.3": Goldener Schnitt

Wir nennen die Zahl $\xi = \frac{1+\sqrt{5}}{2}$ den goldenen Schnitt.

„Vorgehen": Fibonaccisuche

Siehe S. 296 f.

Proposition 7.2.6

a.) Die Fibonaccisuche ist eine Implementierung des allgemeinen Suchverfahrens, also falls

$f(x_k) \geq f(y_k)$ und somit $a_{k+1} = x_k$, so ist $x_{k+1} = y_k$

und falls

$f(x_k) < f(y_k)$ und somit $a_{k+1} = a_k$, so ist $y_{k+1} = x_k$.

b.) Die Fibonaccisuche plaziert x und y symmetrisch in $[a,b]$, d.h für $k < N$ gilt

$$x_k - a_k = b_k - y_k = \frac{F_{N+1-k}}{F_{N+3-k}} (b_k - a_k).$$

c.) Für $k \leq N$ ist $b_k - a_k = \frac{F_{N+3-k}}{F_{N+3}} (b-a)$.

Satz 7.2.7

Seien die S_i wie in Proposition 7.2.1 definiert und setzen wir

$S_{N+1} = y_N - a_N = b_N - x_N$ und

$S_{N+2} = \max\{x_N - a_N, y_N - x_N, b_N - y_N\}$.

Dann gilt

$$S_{N+2} \geq \frac{1}{F_{N+3}}.$$

Also kann kein Suchverfahren bei fest vorgewählter Schrittzahl eine stärkere Reduktion des Suchintervalls garantieren.

„Nachteile der Fibonaccisuche"

a.) Anzahl Iterationen bzw. angestrebende Größe des Suchintervalls muss im Voraus bekannt sein.

b.) Tabelle der Fibonaccizahlen muss bereitgestellt werden

→ In Gleitkommadarstellung angenähert oder Langzahlarithmetik

„Vorgehen": Goldener-Schnitt-Suche

Siehe S. 300 ff.

• Vorschrift Goldener-Schnitt-Suche

$$x_k := a_k + \frac{3-\sqrt{5}}{2}(b_k - a_k)$$

$$y_k := a_k + \frac{\sqrt{5}-1}{2}(b_k - a_k)$$

Algorithmus 7.2.8

```
leftfak = (3-sqrt(5))/2
rightfak = (sqrt(5)-1)/2

def choosepoint(f, a, x, b, fx):
  if b-x <= x-a:
    return a+leftfak*(b-a), x, f(a+leftfak*(b-a)), fx
  else:
    return x, a+rightfak*(b-a), fx, f(a+rightfak*(b-a))

def findmin(f, a, x, b):
  laenge = abs(a) + abs(b)
  fx = f(x)
  while (b-a)/laenge >= eps:
    x, y, fx, fy = choosepoint(f, a, x, b, fx)
    if fx >= fy:
      a = x
      x = y
      fx = fy
    else:
      b=y
else: return (a+b)/2
```

Koordinatensuche und Methode des steilsten Abstiegs

„Verfahren": Koordinatenabstiegsmethode

Sei $\bar{x} \in \mathbb{R}^n$ gegeben. Wir fixieren alle Koordinaten bis auf die i-te und lösen $\min_{x_i \in \mathbb{R}} f(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_{i-1}, x_i, \bar{x}_{i+1}, \dots, \bar{x}_n)$.

Ist x^*_i die optimale Lösung dieses Suchproblems, so setzen wir $\bar{x}_i = x_i^*$, wählen eine andere Koordinate und fahren fort.

„Algorithmus"

```
while ||x - xold|| > ε:
  for i in range(n):
    xold = x
    choose λ ∈ argmin_λ f(x+λe_i)    # löse mit linear search
    x = xold + λe_i
```

Menge aller λ, an denen das Minimum angenommen wird

- obige Methode heißt zyklisches Abstiegsverfahren
- Aitken double sweep method: Koordinaten in Reihenfolge abarbeiten $1, 2, \dots, n-1, n, n-1, \dots, 2, 1, 2, \dots$
- Gauß-Southwell-Verfahren: zusätzlich Differenzierbarkeitsinformationen ausnutzen und stets Koordinate mit dem größten Absolutwert im Gradienten wähle

„Definition": stationärer Punkt

Sei f differenzierbar. Wir nennen einen Punkt, an dem der Gradient verschwindet, einen stationären Punkt.

Satz 7.3.4

Ist $f: \mathbb{R}^n \to \mathbb{R}$ stetig differenzierbar und ist $(x_i)_{i \in N}$ eine Folge, die von einem Koordinatensuchverfahren erzeugt wird, das in jede Koordinatenrichtung unendlich oft sucht, so konvergiert jede konvergente Teilfolge $(x_{i_j})_{j \in N}$ gegen ein x^* mit $\nabla f(x^*) = 0$.

„Methode des steilsten Abstiegs"

Laut Definition 6.4.1 ist eine zulässige Richtung d eine Abstiegsrichtung einer Funktion f, wenn $\nabla f(x) d < 0$.

Mit $\|d\| = 1$ wird diese Zahl betragsmäßig am größten, wenn

$$-d = \frac{\nabla f(x)^T}{\|\nabla f(x)\|}$$ ist.

Methode des steilsten Abstiegs:

```
while ||∇f(x)|| > ε:
  choose λ* ∈ argmin_λ f(x - λ∇f(x)^T)
  x = x - λ* ∇f(x)^T
```

Satz 7.3.5

Sei $f: \mathbb{R}^n \to \mathbb{R}$ stetig differenzierbar und sei $(x_i)_{i \in N}$ eine konvergente Teilfolge einer von der Methode des steilsten Abstiegs erzeugten Punktfolge. Dann konvergiert $(x_i)_{i \in N}$ gegen einen stationären Punkt x^*, d.h. $\nabla f(x^*) = 0$.

Definition 7.3.6 : Konvergenzrate, Konvergenzfaktor und superlineare Konvergenz

Sei $(a_i)_{i \in N}$ eine konvergente Folge reeller Zahlen mit $\lim_{i \to \infty} a_i = a$.

Die Konvergenzrate ist dann das Supremum der nicht negativen Zahlen $p \in \mathbb{R}_+$ mit

$$0 \leq \lim_{i \to \infty} \sup \frac{|a_{i+1} - a|}{|a_i - a|^p} < \infty.$$

Ist p die Konvergenzrate und $1 \leq q \leq p$, so sagen wir auch, die Folge konvergiert von der Ordnung q.

Ist die Konvergenzrate mindestens 2, so sagen wir die Folge konvergiert quadratisch.

Ist die Konvergenzrate mindestens 1, so sagen wir die Folge konvergiert linear mit Konvergenzfaktor κ, wenn

$$\lim_{i \to \infty} \frac{|a_{i+1} - a|}{|a_i - a|} = \kappa < 1.$$

Gilt dies sogar $\kappa = 0$, so sprechen wir von superlinearer Konvergenz.

Bemerkung 7.3.7

In Definition 7.3.7 können undefinierte Ausdrücke auftreten, wenn Folgenglieder gleich dem Grenzwert sind. Gibt es unendlich viele vom Grenzwert verschiedene Folgenglieder, können die undefinierten Ausdrücke ignoriert werden. Bei nur endlich vielen vom Grenzwert verschiedenen Gliedern legen wir die Konvergenzrate auf 0 fest.

Satz 7.3.8

Sei $f: \mathbb{R}^n \to \mathbb{R}$ zweimal stetig differenzierbar und x^* ein relatives Minimum von f. Sei ferner die Hessematrix $\nabla^2 f(x^*)$ positiv definit mit größtem Eigenwert $\lambda_1 > 0$ und kleinstem Eigenwert $\lambda_n > 0$. Ist dann $(x_i)_{i \in \mathbb{N}}$ eine von dem Gradientenabstiegsverfahren erzeugte, gegen x^* konvergente Folge, dann konvergiert die Folge der Zielfunktionswerte $(f(x_i))_{i \in \mathbb{N}}$ linear gegen $f(x^*)$ mit einem Konvergenzfaktor von höchstens $\left(\frac{\lambda_1 - \lambda_n}{\lambda_1 + \lambda_n}\right)$.

Bemerkung 7.3.9

Die Gradienten aufeinanderfolgender Iterationspunkte stehen beim Gradientensuchverfahren senkrecht aufeinander, d.h. es gilt:

$$\nabla f(x_k) \nabla f(x_{k+1})^T = 0.$$

Bemerkung 7.3.10

Bei der Benutzung von Ableitungen in numerischen Algorithmen nähert man diese üblicherweise nur an, d.h. der Gradient $\nabla f(x^*)$ wird etwa angenähert durch den Ausdruck

$$\frac{1}{h}\left(f(x^*+he_1) - f(x^*), \dots, f(x^*+he_n) - f(x^*)\right).$$

Newtonverfahren

„Allgemein"

- Hauptvorteil des Newtonverfahrens: lokales Konvergenzverhalten deutlich besser als bei der Gradientensuche
- Newtonverfahren zur Bestimmung einer Nullstelle aus der Schule:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

Funktion $y = f(x)$ lokal iterativ durch lineare Funktion $\tilde{y}(x) = f(x_k) + (x - x_k) f'(x_k)$ annähern.

Von diesen ausgehend wird als nächster Iterationspunkt x_{k+1} die Nullstellen bestimmt, also

$$0 = f(x_k) + x f'(x_k) - x_k f'(x_k)$$

$$\Leftrightarrow \quad x = x_k - \frac{f(x_k)}{f'(x_k)}.$$

Wir wählen also als nächsten Punkt $x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$.

- Suche nach stationären Punkt statt nach einem lokalen Minimum. Betrachten von f' an Stelle von f im obigen Vorgehen liefert Vorschrift

$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}.$$

Interpretation: f wird lokal durch quadratische Funktion

$$q(x) = f(x_k) + f'(x_k)(x - x_k) + \frac{1}{2} f''(x_k)(x - x_k)^2$$

approximiert.

Für nächsten Iterationspunkt wird der eindeutig stationäre Punkt der quadratischen Funktion berechnet.

„Newton-Verfahren zur nichtlinearen Optimierung"

Sei $f: \mathbb{R}^n \to \mathbb{R}$ eine Funktion.

while $\|x - x_{old}\| > \varepsilon$:

$x_{old} = x$

$x = x_{old} - (\nabla^2 f(x_{old}))^{-1} \nabla f(x_{old})^T$

„mögliche Probleme beim Newtonverfahren"

a.) Im Laufe des Verfahrens kann die Hesse-Matrix singulär oder schlecht konditioniert werden

b.) Es kann passieren, dass $f(x_{k+1}) > f(x_k)$ ist

c.) Die Folge der generierten Punkte kann gegen einen Sattelpunkt konvergieren

Satz 7.4.4

Sei $f: \mathbb{R}^n \to \mathbb{R}$ zweimal stetig differenzierbar und $\nabla f(x^*) = 0$. Sei ferner $\nabla^2 f(x^*)$ regulär und x_1 ein Startpunkt, so dass es $\delta_1, \delta_2 > 0$ gibt mit $\delta_1 \delta_2 < 1$ und für alle x mit $\|x - x^*\| < \|x_1 - x^*\|$ gelte:

a.) $\|(\nabla^2 f(x))^{-1}\|_2 \le \delta_1$

b.) $\|\nabla f(x^*)^T - \nabla f(x)^T - \nabla^2 f(x)(x^* - x)\| \le \delta_2 \|x^* - x\|$.

Dann konvergiert das Newtonverfahren gegen x^*.

Satz 7.4.5

Sei $f: \mathbb{R} \to \mathbb{R}$ viermal stetig differenzierbar, x^* ein stationärer Punkt, also mit $f'(x^*) = 0$ und $f''(x^*) \neq 0$. Sei $(x_k)_{k \in \mathbb{N}}$ eine vom Newtonverfahren erzeugte, gegen x^* konvergente Folge, also

$$x_{k+1} = N(x_k) := x_k - \frac{f'(x_k)}{f''(x_k)}.$$

Dann konvergiert die Folge quadratisch, d.h. mit Konvergenzrate 2.

„Bemerkung"

Newtonverfahren konvergiert nicht unbedingt. Es gibt verschiedene Modifikationen, um Konvergenz zu erzwingen. Skizze siehe 313 f.

Verfahren der konjugierten Richtungen

Seite 314 – 323

Definition 7.5.1: Q-orthogonal, Q-konjugiert, konjugiert

Sei $Q \in \mathbb{R}^{n \times n}$ eine quadratische, symmetrische Matrix. Dann heißen zwei Vektoren $d_1, d_2 \in \mathbb{R}^n$ Q-orthogonal, Q-konjugiert oder kurz konjugiert, wenn $d_1^T Q d_2 = 0$ gilt. Eine endliche Menge von Vektoren $d_1, \dots, d_k \in \mathbb{R}^n$ heißt Q-orthogonal, wenn sie paarweise konjugiert sind.

Proposition 7.5.2

Sei $Q \in \mathbb{R}^{n \times n}$ eine quadratische, symmetrische, positiv definite Matrix und seien $d_1, \dots, d_k \in \mathbb{R}^n \setminus \{0\}$ Q-orthogonal. Dann sind diese Vektoren linear unabhängig.

Satz 7.5.3

Seien Q, b wie eben, $d_1, \dots, d_n$ Q-orthogonal und $x_0 \in \mathbb{R}^n$. Bezeichne B_k den von $d_1, \dots, d_k$ an x_0 aufgespannten affinen Unterraum

$$B_k := \{x_0 + \sum_{j=1}^{k} \lambda_j d_j \mid \lambda_j \in \mathbb{R}\}$$

$$= \{y \in \mathbb{R}^n \mid d_i^T Q (y - x_0) = 0,\ i = k+1, \dots, n\} \quad \text{von } \mathbb{R}^n.$$

Seien nun $x_1, \dots, x_n$ definiert durch

$$x_k := x_{k-1} - \frac{x_{k-1}^T Q d_k - b^T d_k}{d_k^T Q d_k} d_k.$$

Dann ist x_k die Optimallösung des Problems

$$\min_{x \in B_k} \frac{1}{2} x^T Q x - b^T x.$$

Insbesondere löst x_n das volle quadratische Problem.

Methode der konjugierten Gradienten

Sei $x_0 \in \mathbb{R}^n$ und $d_1 = -g_0 = b - Qx_0$.
Wir berechnen iterativ

$$x_k = x_{k-1} - \frac{g_{k-1}^T d_k}{d_k^T Q d_k} \qquad (7.8)$$

$$g_k = Qx_k - b \qquad (7.9)$$

$$d_{k+1} = -g_k + \frac{g_k^T Q d_k}{d_k^T Q d_k}. \qquad (7.10)$$

Definition 7.5.5: Lineare Hülle

Sei $S \subseteq \mathbb{R}^n$. Mit $\text{lin}(S)$ bezeichnen wir die Menge aller Linearkombinationen, die aus Vektoren von S gebildet werden:

$$\text{lin}(S) := \left\{ \sum_{s \in S} \lambda_s s \mid \lambda_s \in \mathbb{R}, \text{ nur endlich viele } \lambda_s \neq 0 \right\}.$$

Wir nennen $\text{lin}(S)$ die lineare Hülle von S.

Satz 7.5.6

Die in (7.8), (7.9), (7.10) definierte Methode der konjugierten Gradienten erfüllt die Voraussetzungen von Satz 7.5.3. Insbesondere gilt, falls das Verfahren nicht in x_k terminiert, dann

a.) für $k = 0, \dots, n-1$: $\text{lin}(\{g_0, g_1, \dots, g_k\}) = \text{lin}(\{g_0, Qg_0, Q^2 g_0, \dots, Q^k g_0\})$.

b.) für $k = 0, \dots, n-1$: $\text{lin}(\{d_1, \dots, d_{k+1}\}) = \text{lin}(\{g_0, Qg_0, Q^2 g_0, \dots, Q^k g_0\})$

c.) $d_{k+1}^T Q d_i = 0$ für $1 \leq i \leq k < n$

d.) $-\frac{g_{k-1}^T d_k}{d_k^T Q d_k} = \frac{g_{k-1}^T g_{k-1}}{d_k^T Q d_k}$

e.) $\frac{g_k^T Q d_k}{d_k^T Q d_k} = \frac{g_k^T g_k}{g_{k-1}^T g_{k-1}}$.

„mögliche Erweiterungen auf nichtquadratische Probleme"

Quadratische Approximation

Sei $x_0 \in \mathbb{R}^n$ und $d_1 = -g_0 = -(\nabla f(x_0))^T$.
while Abbruchbedingung noch nicht erfüllt:
 for $k = 1, \dots, n$:
  $x_k = x_{k-1} - \frac{g_{k-1}^T d_k}{d_k^T \nabla^2 f(x_{k-1}) d_k} d_k$
  $g_k = (\nabla f(x_k))^T$
  if $k \neq n$:
   $d_{k+1} = -g_k + \frac{g_k^T \nabla^2 f(x_k) d_k}{d_k^T \nabla^2 f(x_k) d_k} \cdot d_k$
  else:
   $x_0 = x_n$, $d_1 = -g_0 = -(\nabla f(x_0))^T$.

- Vorteil: kein expliziter line search
- Nachteil: ständige Neuberechnung der Hessematrix sehr aufwändig
  Methode im Allgemeinen nicht global konvergent

Methode nach Fletcher-Reeves

Sei $x_0 \in \mathbb{R}^n$ und $d_1 = -g_0 = -(\nabla f(x_0))^T$

while Abbruchbedingung noch nicht erfüllt:
 for $k = 1, \dots, n$:
  $\alpha_k = \text{argmin } f(x_{k-1} + \alpha d_k)$ # löse mit line search
  $x_k = x_{k-1} + \alpha_k d_k$
  $g_k = (\nabla f(x_k))^T$
  if $k \neq n$:
   $d_{k+1} = -g_k + \frac{g_k^T g_k}{g_{k-1}^T g_{k-1}} d_k$
  else:
   $x_0 = x_n$, $d_1 = -g_0 = -(\nabla f(x_0))^T$.

Satz 7.5.8: Spacer Step Theorem

Seien $X \subseteq \mathbb{R}^n$, $A: X \to X$ eine Funktion und $B: X \to X$ eine stetige Funktion. Sei ferner $f: \mathbb{R}^n \to \mathbb{R}$ eine stetige Funktion und Γ die Menge der stationären Punkte von f. Ferner gelte

$$f(B(x)) < f(x) \text{ für alle } x \in X \setminus \Gamma. \qquad (7.11)$$

Sei nun $(y_n)_{n \in \mathbb{N}}$ eine konvergente Folge mit $y_0 \in X$, $y_{n+1} = A(y_n)$ und $\lim_{n \to \infty} y_n = x^*$.
Sei ferner $\mathcal{K} \subseteq \mathbb{N}$ eine unendliche Indexmenge mit $y_{n+1} = B(y_n)$ falls $n \in \mathcal{K}$.
Dann gilt $x^* \in \Gamma$.

Lineare Optimierung

„Allgemeines lineares Optimierungsproblem"

$\min_{x \in S} c(x)$ mit linearer Funktion $c(x) = c^T x = \sum_{i=1}^n c_i x_i$

und $S \subseteq \mathbb{R}^n$ durch Menge linearer (Un-)Gleichungen gegeben:

$$\sum_{i=1}^n a_i x_i = a^T x \leq \beta$$

$$\sum_{i=1}^n a_i x_i = a^T x \geq \beta \quad \text{bzw.}$$

$$\sum_{i=1}^n a_i x_i = a^T x = \beta$$

Modellbildung

Siehe Seite 340 - 343 für Beispiel

„Bemerkung"

Bei der linearen Optimierung wollen wir anstatt von Minimierungs- lieber Maximierungsprobleme betrachten. Hierbei gilt

$$\max_{x \in S} c(x) = -\left(\min_{x \in S} -c(x)\right).$$

Definition 8.1.4: Lineares Optimierungsproblem / Lineares Programm (LP)

Sei $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $b \geq 0$ und $c \in \mathbb{R}^n$. Die Aufgabenstellung

$$\max c^T x \quad \text{unter } Ax = b,\ x \geq 0$$

nennen wir Lineares Optimierungsproblem in Standardform oder auch Lineares Programm (LP) in Standardform.

Ist $x \geq 0$ mit $Ax = b$, so nennen wir x zulässig für das Problem.

Gibt es einen zulässigen Punkt x, so nennen wir ein lineares Optimierungsproblem zulässig.

Die Menge aller zulässigen Punkte nennen wir den zulässigen Bereich und ein lineares Problem heißt beschränkt, wenn sein zulässiger Bereich eine beschränkte Menge ist.

„Umformung beliebiger Modelle in Standardform"

- $b \geq 0$ kann immer durch Multiplikation mit -1 für die entsprechende Nebenbedingung erreicht werden
- Schlupfvariable z_1

  Für Gleichung $\sum_{i=1}^n a_i x_i = a^T x \leq \beta$ ist Schlupfvariable z_1

  $$z_1 = \beta - \sum_{i=1}^n a_i x_i = \beta - a^T x \geq 0.$$

  Die Ungleichung wird damit zu

  $$\sum_{i=1}^n a_i x_i + z_1 = a^T x + z_1 = \beta, \quad z_1 \geq 0.$$

- Führe für alle Ungleichungen Schlupfvariablen ein und übertrage sie in Matrix A und Vektor b mit $Ax = b$.
- evtl. negative Variablen durch Aufsplitten der Variablen in nicht-negative Variablen überführen:

  Ist u eine nicht vorzeichenbeschränkte Variable, so setze $u = u^+ - u^-$ mit $u^+, u^- \geq 0$ und ersetze u in den Restriktionen und der Zielfunktion (etwa $c_i u$ durch $c_i u^+ - c_i u^-$).

„Bemerkung"

- Simplexalgorithmus ist der älteste und praktisch wichtigste Algorithmus zur Lösung linearer Programme

„Graphische Lösung von Problemen mit zwei Variablen"

Lösungsmenge einer nichttrivialen Gleichung im $\mathbb{R}^n$ ist eine Hyperebene, in der Ebene bekanntlich eine Gerade. Die Ungleichungen definieren dann jeweils einen abgeschlossenen Halbraum, der von so einer Hyperebene berandet wird. Der zulässige Bereich ist also Schnitt von abgeschlossenen Halbräumen, den wir einen Polyeder nennen.

Graphische Methode: Optimallösung erhält man, indem Isoerlösfläche so weit wie möglich in Richtung wachsender Erlöse verschiebt, bis sie das Polyeder nur noch berührt.

Definition 8.1.8: Polyeder

Eine Menge $P \subseteq \mathbb{R}^n$ heißt Polyeder, wenn es ein $m \in \mathbb{N}$, eine Matrix $A \in \mathbb{R}^{m \times n}$ und ein $b \in \mathbb{R}^m$ gibt mit $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$.

Der Dualitätssatz der Linearen Optimierung

Definition 8.2.1: duale Programm

Das Lineare Programm (D) $\min y^T b$ unter $y^T A \geq c^T$ heißt das duale Programm zum Linearen Programm in Standardform.

Lemma 8.2.2: schwache Dualität

Ist $x \geq 0$ zulässig für das primale Programm und y zulässig für das duale Programm, so gilt $c^T x \leq y^T b$.

„Definition": primale Programm

primale Programm (P) $\max c^T x$ unter $Ax = b$, $x \geq 0$.

Satz 8.2.3: Dualitätssatz der Linearen Programmierung

Seien $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ und $c \in \mathbb{R}^n$ und A von vollem Zeilenrang m. Dann gilt:

Ist das primale Programm zulässig und beschränkt, so ist auch das das duale Programm zulässig und beschränkt und es gibt Optimallösungen x^* des primalen bzw. y^* des dualen Programms mit $c^T x^* = y^{*T} b$.

Satz 8.2.4

Seien $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$ und A von vollem Zeilenrang. Dann gilt genau eine der folgenden vier Alternativen:

a.) Das primale und das duale Programm sind zulässig und beschränkt und ihre Zielfunktionswerte sind gleich.

b.) Das primale Programm ist unbeschränkt, d.h. es gibt ein $x_0 \geq 0$ mit $Ax_0 = b$ und ein $x_1 \geq 0$ mit $Ax_1 = 0$ und $c^T x_1 > 0$.

c.) Das duale Programm ist unbeschränkt, d.h. es gibt ein y_0 mit $y_0^T A \geq c^T$ und ein y_1 mit $y_1^T A \geq 0$ und $y_1^T b < 0$.

d.) Beide Programme sind unzulässig, d.h. es gibt weder ein $x \geq 0$ mit $Ax = b$ noch ein $y \in \mathbb{R}^m$ mit $y^T A \geq c^T$.

Satz 8.2.8: Satz vom Komplementären Schlupf

Seien $x^* \in \mathbb{R}^n_+$ mit $Ax^* = b$ und $y^* \in \mathbb{R}^m$ mit $y^{*T} A \geq c^T$. Dann sind x^*, y^* genau dann ein Paar von Optimallösungen des primalen bzw. dualen Programmes, wenn gilt:

a.) $x_i^* \neq 0 \Rightarrow (c^T = y^{*T} A)_i$ und

b.) $(c^T < y^{*T} A)_i \Rightarrow x_i^* = 0$.

Das Simplexverfahren

„Geometrische Idee des Simplexverfahrens"

Gehe solange von einer besseren Ecke, bis es nicht mehr besser geht.

Definition 8.3.1: Ecke eines Polyeders

Sei P ein Polyeder. Dann heißt $x \in P$ Ecke von P, wenn x nicht echte Konvexkombination verschiedener Elemente in P ist, d.h. wenn aus $y, z \in P$ und $x \in \,]y, z[$ notwendig folgt, dass $x = y = z$ ist.

„Notation": $A_{.I}$

Sei I eine Indexmenge und $A \in \mathbb{R}^{m \times n}$ eine Matrix.
Wir bezeichnen mit

a.) $A_{.I}$ die Spalten von A mit Index in I

b.) $A_{I.}$ die Zeilen von A mit Index in I

c.) x_I für ein $x \in \mathbb{R}^n$ der Vektor mit den Einträgen mit Indizes I in x.

Lemma 8.3.2

Habe $A \in \mathbb{R}^{m \times n}$ vollen Zeilenrang m und seien $b \in \mathbb{R}^m$, $x \in \mathbb{R}^n$ mit $x \geq 0$. Dann gilt:

x ist Ecke von $P := \{x \in \mathbb{R}^n \mid Ax = b,\ x \geq 0\}$ $\iff$

es gibt $B \subseteq \{1, \dots, n\}$ mit $A_{.B}$ regulär, $x_B = A_{.B}^{-1} b \geq 0$ und mit $N := \{1, \dots, n\} \setminus B$ gilt $x_N = 0$.

Definition 8.3.3: zulässige Basis (-lösung)

Habe $A \in \mathbb{R}^{m \times n}$ vollen Zeilenrang m und sei $b \in \mathbb{R}^m$. Wir sagen $B \subseteq \{1, \dots, n\}$ ist eine Basis von A, falls $A_{\cdot B}$ regulär ist. Eine Basis B heißt zulässige Basis, wenn darüber hinaus $A_{\cdot B}^{-1} b \geq 0$ ist. Ist B eine (zulässige) Basis, so heißt $N := \{1, \dots, n\} \setminus B$ (zulässige) Nichtbasis und der Vektor $x \in \mathbb{R}^n$ mit $x_B = A_{\cdot B}^{-1} b$, $x_N = 0$ (zulässige) Basislösung.

Proposition 8.3.5: Optimalitätskriterium

Habe $A \in \mathbb{R}^{m \times n}$ vollen Zeilenrang m und sei $b \in \mathbb{R}^m$. Sei B eine zulässige Basis. Dann ist die zulässige Basislösung x eine Optimallösung des linearen Programms, wenn

$$c^T - c_B^T A_{\cdot B}^{-1} A \leq 0.$$

Definition 8.3.6: reduzierten Kosten / Schattenpreise

Ist B eine zulässige Basis, so heißen $c^T - c_B^T A_{\cdot B}^{-1} A$ die reduzierten Kosten oder auch Schattenpreise bzgl. der Basis B.

Definition 8.3.7: benachbarte Basen

Zwei Basen B_1, B_2 heißen benachbart, wenn $|B_1 \cap B_2| = m-1$.

Lemma 8.3.8

Seien B und $B' = (B \cup \{j\}) \setminus \{i\}$ mit $i \in B$, $j \notin B$ zwei benachbarte zulässige Basen mit zugehörigen Basislösungen x und x'. Dann ist

$$c^T x' - c^T x = x'_j (c_j - c_B^T A_{\cdot B}^{-1} A_{\cdot j}).$$

„Bemerkung": S. 358

Proposition 8.3.9: Finde Index des basisverlassenden Elements

Ist B eine zulässige Basis, $i \in B$, $j \notin B$ und

$$i \in \operatorname{argmin} \left\{ \frac{(A_{\cdot B}^{-1} b)_i}{(A_{\cdot B}^{-1} A_{\cdot j})_i} \,\middle|\, (A_{\cdot B}^{-1} A_{\cdot j})_i > 0 \right\},$$

wobei argmin die Menge aller Indizes ist, an denen das Minimum angenommen wird, so ist $B := (B \cup \{j\}) \setminus \{i\}$ eine zulässige Basis.

Lemma 8.3.10

Ist B eine zulässige Basis mit Basislösung x, $c_j - c_B^T A_{\cdot B}^{-1} A_{\cdot j} > 0$ und $(A_{\cdot B}^{-1} A_{\cdot j})_i \leq 0$ für alle i, so ist das lineare Programm unbeschränkt.

Algorithmus 8.3.11: Schematische Skizze des Simplexalgorithmus

Eingabedaten: $A \in \mathbb{R}^{m \times n}$ mit vollem Zeilenrang
$b \in \mathbb{R}^m$ mit $b \geq 0$
zulässige Basis B
$c \in \mathbb{R}^n$

While $c^T - c_B^T A_{\cdot B}^{-1} A \not\leq 0$:

Spaltenwahl: Wähle j mit $c_j - c_B^T A_{\cdot B}^{-1} A_{\cdot j} > 0$.

Zeilenwahl:

Berechne $i \in \operatorname{argmin} \left\{ \frac{(A_{\cdot B}^{-1} b)_i}{(A_{\cdot B}^{-1} A_{\cdot j})_i} \,\middle|\, (A_{\cdot B}^{-1} A_{\cdot j})_i > 0 \right\}$

Falls diese Menge leer ist: STOP. Das Programm ist unbeschränkt.

Basiswechsel: Sei k der Spaltenindex, in dem in $A_{\cdot B}^{-1} A$ der i-te Einheitsvektor steht. Setze $B = (B \cup \{j\}) \setminus \{k\}$.

Tableauform des Simplexalgorithmus

Siehe S. 361 ff.

Pivotwahl, Entartung, Endlichkeit

Pivotregeln (Vorschriften zur Auswahl der Pivotspalte)

Steilster Anstieg

Wähle die Spalte mit den größten reduzierten Kosten.

Größter Fortschritt

Wähle die Spalte, deren Aufnahme in die Basis die größte Verbesserung in der Zielfunktion liefert.

Bland's rule

Wähle stets die Variable mit dem kleinsten Index (sowohl bei Spalten- als auch bei Zeilenwahl). Also wählt man in der Kopfzeile die erste Spalte mit positiven reduzierten Kosten. Falls man bei der Zeilenwahl Alternativen hat, so wählt man diejenige Zeile, bei der das zugehörige Basiselement den kleinsten Spaltenindex hat.

Definition 8.5.2: entartete Ecke

Die Ecke x eines LP in Standardform heißt entartet, wenn es mindestens zwei Basen $B \neq B'$ gibt mit $x_B = A_{\cdot B}^{-1} b$ und $x_{B'} = A_{\cdot B'}^{-1} b$, d.h. die zugehörigen Basislösungen zu B und B' sind gleich. Ebenso nennen wir einen Pivotschritt entartet, der die Basislösung nicht verändert.

Proposition 8.5.3

Ist x eine entartete Ecke, so hat x weniger als m Nichtnulleinträge.

Proposition 8.5.4

a.) Ein Pivotschritt von der Basis B nach B' mit Pivotelement a_{ij} ist genau dann entartet, wenn $(A_{\cdot B}^{-1} b)_i = 0 = (A_{\cdot B'}^{-1} b)_i$ ist.

b.) Ist ein Pivotschritt von Basis B zu B' nicht entartet, so ist $c^T x' = c_{B'}^T A_{\cdot B'}^{-1} b > c_B^T A_{\cdot B}^{-1} b = c^T x$.

„Definition": zykeln

Es gibt eine Folge von Basen $B_1, \dots, B_k = B_1$, so dass der Simplexalgorithmus unter Anwendung der Steilsten-Anstieg-Regel von B_i nach B_{i+1} wechselt.

Satz 8.5.6

Bei Anwendung von Bland's rule zykelt das Simplexverfahren nicht.

Satz 8.5.7: Korrektheit des Simplexverfahrens

a.) Bei Anwendung von Bland's rule stoppt das Simplexverfahren nach endlich vielen Schritten.

b.) Wenn das Simplexverfahren stoppt und $c^T - c_B^T A_{\cdot B}^{-1} A \not\leq 0$ ist, so ist das Problem unbeschränkt, andernfalls ist B eine optimale Basis.