Woche 11

Freitag, 19. Mai 2023 09:47

Zerlegungen

1) LU

Sei $A \in GL_n(\mathbb{R})$. Dann existieren

- P: Permutationsmatrix
- L: untere Δ-Matrix
- U: obere Δ-Matrix

} $n \times n$-Matrizen

s.d. $A = PLU$

Sinnvoll, da dann schnelles lösen von LGS mittels Substitution

$$Ax = b \iff L\underbrace{Ux}_{z} = \underbrace{P^T b}_{\tilde{b}}$$

a) löse zuerst $Lz = \tilde{b}$ (Vorwärtssub.)

b) löse dann $Ux = z$ (Rückwärtssub.)

Super, da $\begin{pmatrix} \diagdown & 0 \end{pmatrix} z = \tilde{b} \iff z_1 = \frac{\tilde{b}_1}{L_{11}}$, $z_2 = \frac{\tilde{b}_2 - L_{21} z_2}{L_{22}}$, $\vdots$ (Lösung direkt ausrechnen)

+ existieren sehr stabile Algorithmen für die LU-Zerlegung

2) Cholesky

Falls $A \in GL_n(\mathbb{R})$ symmetrisch + pos. definit, dann existieren

L (untere Δ-Mat.), U (obere Δ-Mat.) s.d. $A = LL^T = UU^T$

3) QR-Zerlegung (orthogonal ≡ bzgl. euklid. Skalarprod.)

Sei $A \in Mat_{m \times n}(\mathbb{R})$ mit vollem Rang. Dann kann man folgende Zerlegungen machen:

1) $m = n \Rightarrow A \in GL_n(\mathbb{R})$

$Q \in O(n)$ $(Q^T Q = 1)$, $R \in GL_n(\mathbb{R})$ (obere Δ-Mat.)

$A = QR$

2) $m > n$:

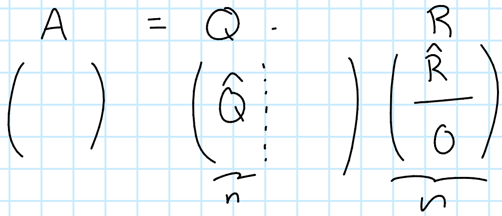
reduziert: $\tilde{Q} \in Mat_{m \times n}(\mathbb{R})$ mit orth. Spalten

$\tilde{R} \in GL_n(\mathbb{R})$ obere Δ-Mat. s.d.

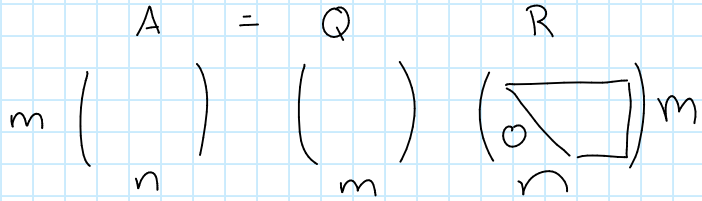
$A = \tilde{Q}\tilde{R}$

vollständig: $Q \in O(n)$ mit $Q = (\hat{Q} \mid q_{n+1} \mid \dots \mid q_m)$

$R \in Mat_{m\times n}(\mathbb{R})$ mit $R = \begin{pmatrix} \hat{R} \\ 0 \end{pmatrix} \begin{matrix} \}n \\ \}m-n \end{matrix}$



3) $m < n$: $Q \in O(n)$, $R \in Mat_{m\times n}(\mathbb{R})$ mit vollem Rang s.d. $A = QR$



Wieso QR-Zerlegung?

Die orthogonalen Matrizen Q sind gut konditioniert, da sie die 2-Norm erhalten: $\|Qx\|_2 = \|x\| \quad \forall Q \in O(n)$

+ orthogonale Koordinaten vereinfachen viele Rechnungen (z.B. Symmetrien in Physik)

Lösen von LGS:

$Ax = b \iff QRx = b$

$\iff Rx = Q^T b$

Verfahren für QR-Zerlegung:

a) Gram-Schmit (GS)

b) Givens-Rot

c) Housholder-Spiegelungen

a) GS

Nehme Spalten von $A = (a_1 | \dots | a_n)$ als Startvektoren für GS-Verfahren

(+) analytisch gut verständlich

Wenn Verfahren in der Mitte abgebrochen wird, dann haben wir trotzdem schon die orth. Vektoren q_i bis dahin

(-) numerisch instabil

Verbesserung: modifiziertes GS - Verfahren
(berücksichtigt Rundungsfehler in jedem Schritt)

Bemerkung: QR-Zerlegung durch Multiplikation mit oberen Δ-Mat erzeugt
⇒ Δ-Mat. nicht gut konditioniert ⇒ Stabilität leidet

Besser: Erstelle Q, indem nur orthogonale Transformationen angewendet werden müssen
↳ Householder / Givens

Idee Householder: Spiegele eine Spalte von A direkt so, dass die gewünschte Spalte von R entsteht

SVD

Hilfreich für Datenkompression, Datenanalyse, Ausgleichsrechnung

Sei $A \in \mathbb{C}^{m\times n}$, dann $\exists V \in U(m)\ \exists U \in U(n)$, $\Sigma := \mathrm{diag}(\sigma_1, \dots, \sigma_p)$ mit $p = \min\{n,m\}$

$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_p \geq 0$ s.d. $A = U\Sigma V^*$

Numerisch: Unterscheidung von numerischem und theoretischen Rang

Durch Rundungsfehler: $\underbrace{\sigma_1 \geq \sigma_2 \geq \dots \sigma_r}_{\text{numerischer Rang}} >> \underbrace{\sigma_{r+1} \approx \dots \approx \sigma_p \approx 0}_{\text{zählt nicht zum Rang}}$

SVD und Kondition:

Satz: Sei $A \in \mathbb{C}^{n\times n}$, dann $\mathrm{cond}_2(A) = \frac{\sigma_1}{\sigma_m}$ (Kondition mit Singulärwerten verbinden)

Anwendungen SVD:

1) Datenkompression: $A \in \mathbb{C}^{m\times n}$

A normal speichern: $m \cdot n$ Werte

$A = \sum_{k=1}^{p} \sigma_k u_k v_k^*$ speichern: $p(1+n+m)$ Werte

⇒ für p klein weniger Speicherplatz (z.B. in Bildkompression)

2) Datenanalyse:

Mit Principal component Analysis kann man Datensets auf die grössten Singulärwerte reduzieren