

Lineare Algebra Woche 4

Orthogonalität, QR-Zerlegung und DCT

Themen: Skalarprodukte, Gram-Schmidt, QR-Zerlegung, Orthogonale Matrizen, DCT/JPEG

Teil I: Skalarprodukte und Orthogonalität

- Allgemeine Skalarprodukte
- Orthogonalbasen

Teil II: Gram-Schmidt-Verfahren

- Orthogonalisierung
- QR-Zerlegung
- Numerische Aspekte

Lernziele: Orthogonalität verstehen und anwenden können

Teil III: Orthogonale Matrizen

- Eigenschaften
- Rotationen und Spiegelungen
- Householder und Givens

Teil IV: DCT und JPEG

- Diskrete Cosinus-Transformation
- Bildkompression
- Praktische Anwendung

Lernziele: Orthogonalität verstehen und anwenden können

Einführung: Grundlagen für Anfänger

Was ist ein Vektor?

Ein Vektor ist ein Pfeil mit:

- Richtung
- Länge
- Komponenten (x, y)

Beispiel:

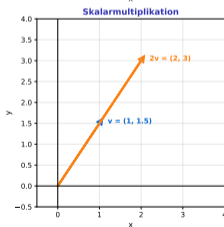
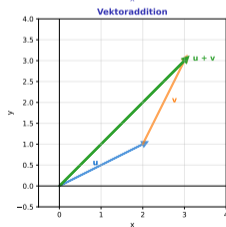
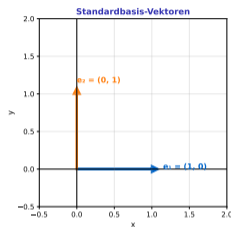
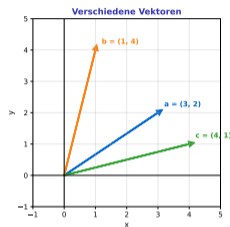
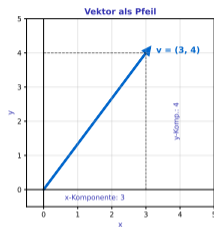
$$v = \begin{pmatrix} 3 \\ 2 \end{pmatrix}$$

- 3 Schritte nach rechts
- 2 Schritte nach oben

Vektoren starten am Ursprung $(0,0)$ und zeigen zu einem Punkt (x,y)

Vektoren in \mathbb{R}^2 : Geometrische Darstellung

Vektoren in \mathbb{R}^2 : Pfeile mit Richtung und Länge



Eigenschaften von Vektoren

- Notation:** $v = (v_1, v_2)$ in \mathbb{R}^2
 $v = (v_1, v_2, v_3)$ in \mathbb{R}^3
- Darstellung:** Als Pfeil mit Richtung und Länge
- Komponente:** Koordinaten entlang der Achsen
- Addition:** Komponenten-weise addieren
 $u + v = (u_1 + v_1, u_2 + v_2)$
- Multiplikation:** $av = (av_1, av_2)$

Wichtig: Vektoren beginnen immer am Ursprung $(0,0)$

Vektoren starten am Ursprung $(0,0)$ und zeigen zu einem Punkt (x,y)

Wann sind Vektoren senkrecht?

Zwei Vektoren sind senkrecht (orthogonal), wenn sie einen 90° -Winkel bilden.

Beispiele:

- Koordinatenachsen: x -Achse \perp y -Achse
- Wände eines Raums
- Ecken eines Rechtecks

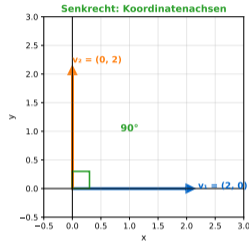
Wichtig:

Senkrechte Vektoren sind unabhängig voneinander!

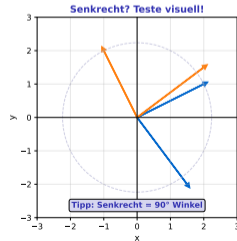
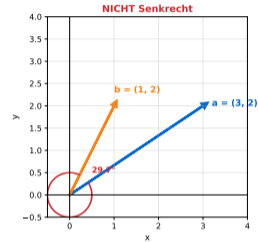
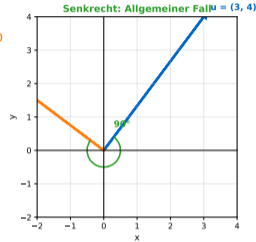
Orthogonale Vektoren sind die Basis für viele Anwendungen in der Linearen Algebra

Senkrechte Vektoren: 90°-Winkel im Koordinatensystem

Senkrechte Vektoren: Der rechte Winkel (90°)



$v = (-4, 3)$



Was bedeutet "senkrecht"?

- Definition:** Zwei Vektoren sind senkrecht, wenn sie einen 90°-Winkel bilden
- Andere Namen:** Orthogonal
- Perpendicular
 - Rechtwinklig
- Beispiele:**
- Koordinatenachsen: $x \perp y$
 - Ecken eines Rechtecks
 - $(3,4) \perp (-4,3)$
- Frage:** Wie prüfen wir das algebraisch?
- Skalarprodukt (nächste Folie!)

Das Skalarprodukt

Für zwei Vektoren $u = (u_1, u_2)$ und $v = (v_1, v_2)$:

$$u \cdot v = u_1 \cdot v_1 + u_2 \cdot v_2$$

Beispiel:

$$\begin{pmatrix} 3 \\ 2 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 4 \end{pmatrix} = 3 \cdot 1 + 2 \cdot 4 = 11$$

Bedeutung:

- Misst, wie parallel zwei Vektoren sind
- Ergebnis = 0 \rightarrow Vektoren senkrecht!

Das Skalarprodukt ist der wichtigste Test für Orthogonalität

Skalarprodukt (Dot Product): Formel und geometrische Bedeutung

Formel: Skalarprodukt

In \mathbb{R}^2 :

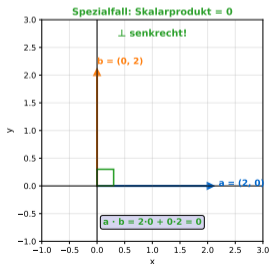
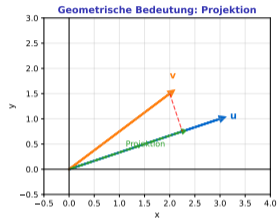
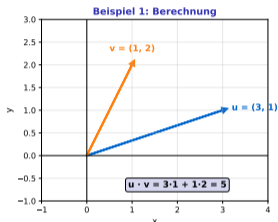
$$\mathbf{u} \cdot \mathbf{v} = u_1v_1 + u_2v_2$$

In \mathbb{R}^3 :

$$\mathbf{u} \cdot \mathbf{v} = u_1v_1 + u_2v_2 + u_3v_3$$

Eigenschaften:

- Ergebnis ist eine Zahl (Skalar)
- $\mathbf{u} \cdot \mathbf{v} = \mathbf{v} \cdot \mathbf{u}$ (kommutativ)
- $\mathbf{u} \cdot \mathbf{u} = \|\mathbf{u}\|^2$ (Länge im Quadrat)



Weitere Beispiele

Vektoren	Rechnung	Ergebnis
$(1, 0) \cdot (1, 0)$	$= 1 \cdot 1 + 0 \cdot 0$	$= 1$
$(3, 4) \cdot (1, 2)$	$= 3 \cdot 1 + 4 \cdot 2$	$= 11$
$(2, 1) \cdot (-1, 2)$	$= 2 \cdot (-1) + 1 \cdot 2$	$= 0$
$(-1, 3) \cdot (3, 1)$	$= -1 \cdot 3 + 3 \cdot 1$	$= 0$
$(1, 1) \cdot (1, 1)$	$= 1 \cdot 1 + 1 \cdot 1$	$= 2$
$(5, 0) \cdot (0, 7)$	$= 5 \cdot 0 + 0 \cdot 7$	$= 0$

Grün = 0 → Vektoren sind senkrecht!

Zusammenfassung

Formel: $\mathbf{u} \cdot \mathbf{v} = \sum u_i v_i$

Ergebnis: Eine Zahl (Skalar)

Bedeutung: Projektion \times Länge

$\mathbf{u} \cdot \mathbf{v} = \|\mathbf{u}\| \|\mathbf{v}\| \cos \theta$

Wichtig!

$\mathbf{u} \cdot \mathbf{v} = 0 \iff \mathbf{u} \perp \mathbf{v}$

Das Skalarprodukt testet Orthogonalität!
(senkrecht / orthogonal)

Die goldene Regel

$$u \perp v \Leftrightarrow u \cdot v = 0$$

Schritte:

1. Berechne Skalarprodukt $u \cdot v$
2. Ist Ergebnis = 0?
 - Ja \rightarrow orthogonal
 - Nein \rightarrow nicht orthogonal

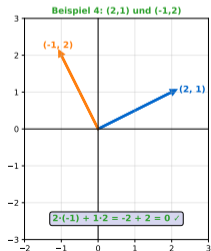
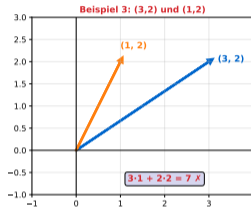
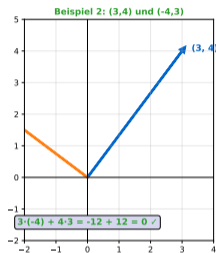
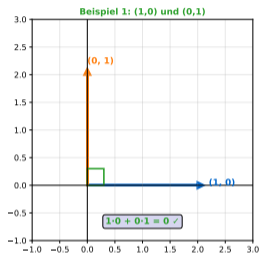
Beispiel:

$$(3, 4) \cdot (-4, 3) = -12 + 12 = 0$$

Test auf Orthogonalität: Nur Skalarprodukt berechnen und prüfen ob = 0

Orthogonalitätstest: Beispiele mit Skalarprodukt = 0

Wann sind Vektoren senkrecht? Test mit Skalarprodukt



Orthogonalitätstest

Regel: Zwei Vektoren sind orthogonal

$$\iff u \cdot v = 0$$

Schritte:

1. Berechne Skalarprodukt $u \cdot v$
2. Prüfe: Ist das Ergebnis 0?
3. Ja \rightarrow orthogonal \checkmark
Nein \rightarrow nicht orthogonal \times

Übersicht: Mehr Beispiele

Vektor u	Vektor v	$u \cdot v$	Status
(1,0)	(0,1)	0	\checkmark
(3,4)	(-4,3)	0	\checkmark
(2,1)	(-1,2)	0	\checkmark
(5,0)	(0,7)	0	\checkmark
(1,1)	(1,1)	2	\times
(3,2)	(1,2)	7	\times

Länge eines Vektors

Die Länge (Betrag) eines Vektors $v = (v_1, v_2)$:

$$\|v\| = \sqrt{v_1^2 + v_2^2}$$

Beispiel:

Für $v = (3, 4)$:

$$\|v\| = \sqrt{3^2 + 4^2} = \sqrt{9 + 16} = \sqrt{25} = 5$$

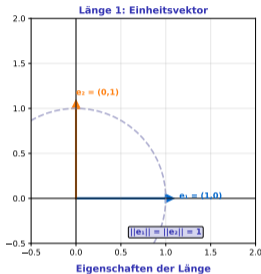
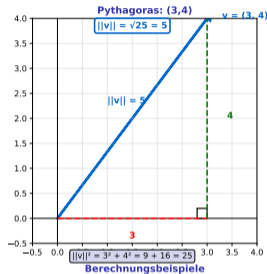
Mit Skalarprodukt:

$$\|v\| = \sqrt{v \cdot v}$$

Die Vektorlänge folgt direkt aus dem Satz des Pythagoras

Vektorlänge: Pythagoras-Satz Darstellung

Vektorlänge (Betrag): $\|v\| = \sqrt{v \cdot v} = \sqrt{v_1^2 + v_2^2}$



Formel: Vektorlänge

In \mathbb{R}^2 :
 $\|v\| = \sqrt{v_1^2 + v_2^2}$

In \mathbb{R}^3 :
 $\|v\| = \sqrt{v_1^2 + v_2^2 + v_3^2}$

Mit Skalarprodukt:
 $\|v\| = \sqrt{v \cdot v}$

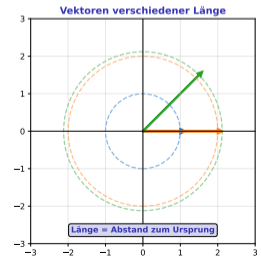
Vektor v	Berechnung	$\ v\ $
$(1, 0)$	$\sqrt{1^2 + 0^2}$	$= 1$
$(0, 1)$	$\sqrt{0^2 + 1^2}$	$= 1$
$(3, 4)$	$\sqrt{3^2 + 4^2}$	$= \sqrt{25} = 5$
$(1, 1)$	$\sqrt{1^2 + 1^2}$	$= \sqrt{2} = 1.41$
$(5, 12)$	$\sqrt{5^2 + 12^2}$	$= \sqrt{169} = 13$
$(6, 8)$	$\sqrt{6^2 + 8^2}$	$= \sqrt{100} = 10$

Positiv: $\|v\| \geq 0$ für alle v
 $\|v\| = 0 \Leftrightarrow v = 0$

Skalierung: $\|a \cdot v\| = |a| \cdot \|v\|$

Dreiecksungleichung: $\|u + v\| \leq \|u\| + \|v\|$

Pythagoras: $u \perp v \Rightarrow \|u+v\|^2 = \|u\|^2 + \|v\|^2$



Was ist Normierung?

Einen Vektor auf Länge 1 bringen:

$$\hat{u} = \frac{v}{\|v\|}$$

Beispiel:

Für $v = (3, 4)$ mit $\|v\| = 5$:

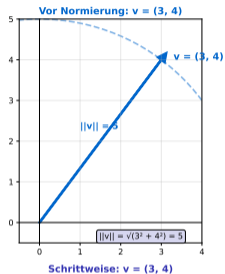
$$\hat{u} = \frac{1}{5} \begin{pmatrix} 3 \\ 4 \end{pmatrix} = \begin{pmatrix} 0.6 \\ 0.8 \end{pmatrix}$$

Eigenschaften:

- Gleiche Richtung wie v
- Länge = 1
- Liegt auf Einheitskreis

Normierte Vektoren (Einheitsvektoren) haben Länge 1 und sind essentiell für orthonormale Basen

Normierung: Einheitsvektoren erstellen ($||\hat{u}|| = 1$)



Schritt 1: Berechne Länge $||v||$

$$||v|| = \sqrt{3^2 + 4^2} = \sqrt{25} = 5$$

Schritt 2: Dividiere durch Länge

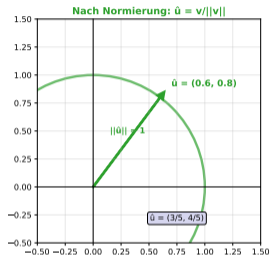
$$\hat{u}_1 = v_1/||v|| = 3/5 = 0.6$$

$$\hat{u}_2 = v_2/||v|| = 4/5 = 0.8$$

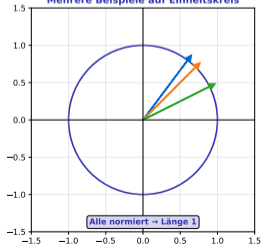
Schritt 3: Ergebnis

$$\hat{u} = (0.6, 0.8)$$

Prüfung: $||\hat{u}|| = \sqrt{0.6^2 + 0.8^2} = 1 \checkmark$



Mehrere Beispiele auf Einheitskreis



Formel: Normierung

Normierungsformel:

$$\hat{u} = v / ||v||$$

Komponentenweise:

$$\hat{u} = (v_1/||v||, v_2/||v||)$$

Ergebnis: $||\hat{u}|| = 1$

Eigenschaften

Länge: $||\hat{u}|| = 1$ immer

Richtung: Gleiche Richtung wie v

Nur Skalierung: $||v||$ (nur Länge ändert sich)

Einheitsvektor: "unit vector" genannt

Verwendung: orthonormale Basen

Zwei wichtige Begriffe

Orthogonal:

Vektoren sind senkrecht

$$u \cdot v = 0$$

(beliebige Länge!)

Orthonormal:

Senkrecht UND Länge 1

$$u \cdot v = 0 \text{ und } \|u\| = \|v\| = 1$$

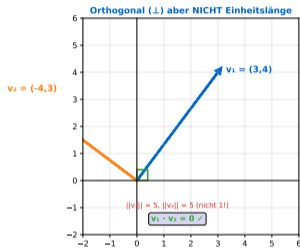
Merke:

Orthonormal = Orthogonal + Normiert

Orthonormale Basen sind ideal für Berechnungen - das Ziel des Gram-Schmidt-Verfahrens

Orthogonal vs. Orthonormal: Visuelle Unterscheidung

Orthogonal vs. Orthonormal: Der Unterschied

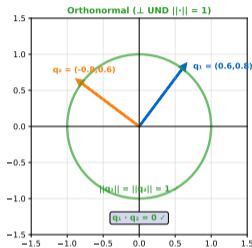


Definitionen

Orthogonal:
 $u \cdot v = 0$
 (senkrecht, beliebige Länge)

Orthonormal:
 $u \cdot v = 0$ UND $\|u\| = \|v\| = 1$
 (senkrecht UND Einheitslänge)

Orthonormal = Orthogonal + Normiert



Von Orthogonal zu Orthonormal

Gegeben: Orthogonale Vektoren
 $v_1 = (3, 4), v_2 = (-4, 3)$
 $v_1 \cdot v_2 = 0 \checkmark$ aber $\|v_1\| = 5 \neq 1$

↓ Normierung

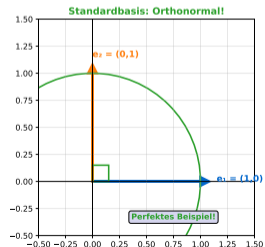
$q_1 = v_1 / \|v_1\| = (3,4)/5 = (0.6, 0.8)$
 $q_2 = v_2 / \|v_2\| = (-4,3)/5 = (-0.8, 0.6)$

↓

Ergebnis: Orthonormale Vektoren
 $q_1 \cdot q_2 = 0 \checkmark$ und $\|q_1\| = \|q_2\| = 1 \checkmark$

Vergleich

Eigenschaft	Orthogonal	Orthonormal
Senkrecht?	✓ Ja	✓ Ja
$u \cdot v = 0$?	✓ Ja	✓ Ja
$\ u\ = 1$?	✗ Nein	✓ Ja
$\ v\ = 1$?	✗ Nein	✓ Ja
Beispiel:	$(3,4) \perp (-4,3)$ $(0.6,0.8) \perp (-0.8,0.6)$	



Aufgabe:

Gegeben: $v_1 = (3, 4)$, $v_2 = (-4, 3)$

Erstelle Orthonormalbasis!

Schritt 1: Orthogonal?

$$v_1 \cdot v_2 = 3(-4) + 4(3) = 0$$

Schritt 2: Einheitslänge?

$$\|v_1\| = \sqrt{9 + 16} = 5 \neq 1$$

Schritt 3: Normieren

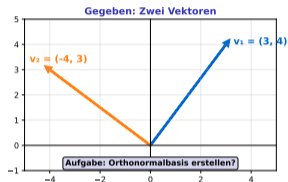
$$q_1 = \frac{v_1}{5} = (0.6, 0.8)$$

$$q_2 = \frac{v_2}{5} = (-0.8, 0.6)$$

Systematisches Vorgehen: Orthogonalität prüfen \rightarrow Länge prüfen \rightarrow Normieren

Vollständiges Beispiel in \mathbb{R}^2 : Schritt-für-Schritt

Vollständiges Beispiel in \mathbb{R}^2 : Von Vektoren zu Orthonormalbasis



Schritt 1: Orthogonal?

Prüfe: $v_1 \cdot v_2 = 0?$

Berechnung:

$$\begin{aligned}v_1 \cdot v_2 &= (3) \cdot (-4) + (4) \cdot (3) \\ &= -12 + 12 \\ &= 0\end{aligned}$$

Ergebnis: Orthogonal ✓

Die Vektoren stehen senkrecht!

Schritt 2: Einheitslänge?

Prüfe: $\|v_1\| = \|v_2\| = 1?$

Berechnung $\|v_1\|$:

$$\|v_1\| = \sqrt{3^2 + 4^2} = \sqrt{25} = 5$$

Berechnung $\|v_2\|$:

$$\|v_2\| = \sqrt{(-4)^2 + 3^2} = \sqrt{25} = 5$$

Ergebnis: NICHT Einheitslänge ✗

$$\|v_1\| = \|v_2\| = 5 \neq 1$$

→ Normierung nötig!

Schritt 3: Normierung

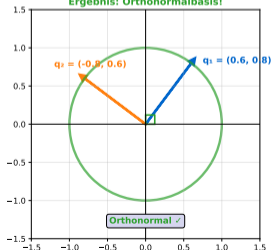
Normiere: $q = v/\|v\|$

$$\begin{aligned}q_1 &= v_1/\|v_1\| = (3, 4)/5 \\ &= (3/5, 4/5) \\ &= (0.6, 0.8)\end{aligned}$$

$$\begin{aligned}q_2 &= v_2/\|v_2\| = (-4, 3)/5 \\ &= (-4/5, 3/5) \\ &= (-0.8, 0.6)\end{aligned}$$

Normierte Vektoren erstellt!

Ergebnis: Orthonormalbasis!



Verifikation

Prüfe alle Bedingungen:

$$q_1 \cdot q_2 = 0?$$

$$q_1 \cdot q_2 = (0.6)(-0.8) + (0.8)(0.6) = 0 \checkmark$$

$$\|q_1\| = 1?$$

$$\|q_1\| = \sqrt{0.6^2 + 0.8^2} = \sqrt{1} = 1 \checkmark$$

$$\|q_2\| = 1?$$

$$\|q_2\| = \sqrt{0.8^2 + 0.6^2} = \sqrt{1} = 1 \checkmark$$

✓ Alle Bedingungen erfüllt!

Die Standardbasis

$$e_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \quad e_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \quad e_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

Orthogonalität:

$$e_1 \cdot e_2 = e_1 \cdot e_3 = e_2 \cdot e_3 = 0$$

Normierung:

$$\|e_1\| = \|e_2\| = \|e_3\| = 1$$

Fazit:

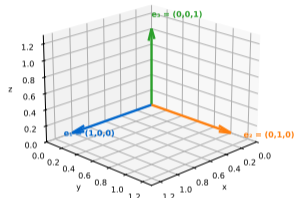
Die Standardbasis ist bereits orthonormal!

Die Standardbasis ist das perfekte Beispiel einer orthonormalen Basis

Vollständiges Beispiel in \mathbb{R}^3 : 3D-Darstellung

Vollständiges Beispiel in \mathbb{R}^3 : Standardbasis ist Orthonormalbasis

Gegeben: Standardbasis in \mathbb{R}^3



Schritt 3: Normierung?

Normierung nötig?

Nein! Bereits normiert:

$$\|e_1\| = 1 \checkmark$$

$$\|e_2\| = 1 \checkmark$$

$$\|e_3\| = 1 \checkmark$$

Die Standardbasis ist bereits

eine Orthonormalbasis!

Schritt 1: Orthogonal?

Prüfe alle Paare: $e_i \cdot e_j = 0$?

$$e_1 \cdot e_2:$$

$$(1) \cdot (0) + (0) \cdot (1) + (0) \cdot (0) = 0 \checkmark$$

$$e_1 \cdot e_3:$$

$$(1) \cdot (0) + (0) \cdot (0) + (0) \cdot (1) = 0 \checkmark$$

$$e_2 \cdot e_3:$$

$$(0) \cdot (0) + (1) \cdot (0) + (0) \cdot (1) = 0 \checkmark$$

Ergebnis: Alle Paare orthogonal!

Ergebnis: Orthonormalbasis!

Schritt 2: Einheitslänge?

Prüfe: $\|e_1\| = \|e_2\| = \|e_3\| = 1$?

Berechnung $\|e_1\|$:

$$\|e_1\| = \sqrt{1^2 + 0^2 + 0^2} = \sqrt{1} = 1 \checkmark$$

Berechnung $\|e_2\|$:

$$\|e_2\| = \sqrt{0^2 + 1^2 + 0^2} = \sqrt{1} = 1 \checkmark$$

Berechnung $\|e_3\|$:

$$\|e_3\| = \sqrt{0^2 + 0^2 + 1^2} = \sqrt{1} = 1 \checkmark$$

Ergebnis: Alle Einheitslänge!

Verifikation: Orthonormalbasis

Alle Bedingungen erfüllt:

Orthogonalität:

$$e_1 \cdot e_2 = e_1 \cdot e_3 = e_2 \cdot e_3 = 0 \checkmark$$

Normierung:

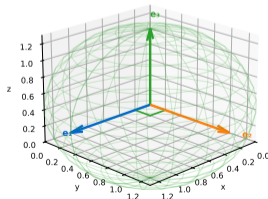
$$\|e_1\| = \|e_2\| = \|e_3\| = 1 \checkmark$$

Fazit:

$\{e_1, e_2, e_3\}$ ist Orthonormalbasis \checkmark

Die Standardbasis in \mathbb{R}^3 ist

das perfekte Beispiel!



Was wir gelernt haben:

1. Vektoren als Pfeile verstehen
2. Senkrechte Vektoren erkennen
3. Skalarprodukt berechnen
4. Orthogonalität testen: $u \cdot v = 0$
5. Vektorlänge berechnen: $\|v\| = \sqrt{v \cdot v}$
6. Vektoren normieren: $\hat{u} = v/\|v\|$
7. Unterschied: Orthogonal vs. Orthonormal
8. Vollständige Beispiele in \mathbb{R}^2 und \mathbb{R}^3

Mit diesen Grundlagen verstehen Sie jetzt die fortgeschrittenen Konzepte der Orthogonalität

Was kommt als Nächstes?

Jetzt können wir verstehen:

- Allgemeine Skalarprodukte
- Gram-Schmidt-Verfahren
- QR-Zerlegung
- Orthogonale Matrizen
- Anwendungen (JPEG/DCT)

Bereit für Teil II!

Mit diesen Grundlagen verstehen Sie jetzt die fortgeschrittenen Konzepte der Orthogonalität

BSc-Level: Orthogonalität und QR-Zerlegung

Warum brauchen wir Gram-Schmidt?

Orthonormalbasen haben viele Vorteile:

- Berechnungen werden einfacher
- Numerisch stabil
- Geometrisch anschaulich

Problem: Natürlich vorkommende Vektoren sind meist *nicht* orthogonal

Beispiel: $a_1 = (1, 1)$, $a_2 = (1, 2)$ sind linear unabhängig, aber $a_1 \cdot a_2 = 3 \neq 0$

Lösung: Gram-Schmidt-Verfahren

- Systematischer Algorithmus für \mathbb{R}^n
- Grundlage für QR-Zerlegung

Gram-Schmidt transformiert beliebige Basis in Orthonormalbasis

Input: Linear unabhängige Vektoren $\{a_1, \dots, a_n\}$

Output: Orthonormalbasis $\{q_1, \dots, q_n\}$

Algorithmus:

1. **Erster Vektor:** Normiere a_1

$$q_1 = \frac{a_1}{\|a_1\|}$$

2. **Weitere Vektoren** (für $i = 2, \dots, n$):

- Subtrahiere Projektionen auf alle vorherigen q_j :

$$v_i = a_i - \sum_{j=1}^{i-1} (a_i \cdot q_j) q_j$$

- Normiere das Ergebnis:

$$q_i = \frac{v_i}{\|v_i\|}$$

Die Idee: Mache jeden neuen Vektor orthogonal zu allen vorherigen

Was macht der Algorithmus?

Für jeden Vektor a_i :

1. Projiziere a_i auf $\text{span}\{q_1, \dots, q_{i-1}\}$
2. Subtrahere diese Projektion von a_i
3. Was übrig bleibt, ist orthogonal zu allen q_1, \dots, q_{i-1} !

Projektion von a_i auf q_j :

$$\text{proj}_{q_j}(a_i) = (a_i \cdot q_j)q_j$$

Orthogonale Komponente:

$$v_i = a_i - \sum_{j=1}^{i-1} \text{proj}_{q_j}(a_i)$$

Wir "räumen" die Komponenten parallel zu den vorherigen Vektoren weg

Gegeben:

$$a_1 = \begin{pmatrix} 3 \\ 4 \end{pmatrix}, \quad a_2 = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

Schritt 1: Normiere a_1

$$\|a_1\| = \sqrt{9 + 16} = 5$$

$$q_1 = \frac{1}{5} \begin{pmatrix} 3 \\ 4 \end{pmatrix} = \begin{pmatrix} 0.6 \\ 0.8 \end{pmatrix}$$

Schritt 2: Orthogonalisiere a_2

$$a_2 \cdot q_1 = 1(0.6) + 2(0.8) = 2.2$$

$$v_2 = a_2 - (a_2 \cdot q_1)q_1 = \begin{pmatrix} 1 \\ 2 \end{pmatrix} - 2.2 \begin{pmatrix} 0.6 \\ 0.8 \end{pmatrix} = \begin{pmatrix} -0.32 \\ 0.24 \end{pmatrix}$$

Fortsetzung auf nächster Folie

Schritt 3: Normiere v_2

$$\|v_2\| = \sqrt{0.32^2 + 0.24^2} = 0.4$$

$$q_2 = \frac{1}{0.4} \begin{pmatrix} -0.32 \\ 0.24 \end{pmatrix} = \begin{pmatrix} -0.8 \\ 0.6 \end{pmatrix}$$

Ergebnis:

$$q_1 = \begin{pmatrix} 0.6 \\ 0.8 \end{pmatrix}, \quad q_2 = \begin{pmatrix} -0.8 \\ 0.6 \end{pmatrix}$$

Überprüfung:

- Orthogonal? $q_1 \cdot q_2 = (0.6)(-0.8) + (0.8)(0.6) = -0.48 + 0.48 = 0 \checkmark$
- Länge 1? $\|q_1\| = \|q_2\| = 1 \checkmark$

Wir haben eine Orthonormalbasis des \mathbb{R}^2 erstellt!

Aufgabe

Wenden Sie Gram-Schmidt an auf:

$$a_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad a_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Schritte:

1. Berechnen Sie $\|a_1\|$ und q_1
2. Berechnen Sie $a_2 \cdot q_1$
3. Berechnen Sie $v_2 = a_2 - (a_2 \cdot q_1)q_1$
4. Berechnen Sie $\|v_2\|$ und q_2
5. Überprüfen Sie: $q_1 \cdot q_2 = 0$ und $\|q_1\| = \|q_2\| = 1$

Lösung auf nächster Folie

Versuchen Sie es selbst, bevor Sie weiterblättern!

Schritt 1:

$$\|a_1\| = \sqrt{1^2 + 1^2} = \sqrt{2}, \quad q_1 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

Schritt 2:

$$a_2 \cdot q_1 = 0 \cdot \frac{1}{\sqrt{2}} + 1 \cdot \frac{1}{\sqrt{2}} = \frac{1}{\sqrt{2}}$$

Schritt 3:

$$v_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix} - \frac{1}{\sqrt{2}} \cdot \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} -1/2 \\ 1/2 \end{pmatrix}$$

Schritt 4:

$$\|v_2\| = \frac{1}{\sqrt{2}}, \quad q_2 = \frac{1}{\sqrt{2}} \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

Ergebnis: $q_1 = (1/\sqrt{2}, 1/\sqrt{2})$, $q_2 = (-1/\sqrt{2}, 1/\sqrt{2})$

Gegeben (nur erste zwei Vektoren):

$$a_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \quad a_2 = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

Schritt 1:

$$\|a_1\| = \sqrt{1+0+0+1} = \sqrt{2}, \quad q_1 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

Schritt 2:

$$a_2 \cdot q_1 = \frac{1}{\sqrt{2}}(1+0+0+0) = \frac{1}{\sqrt{2}}$$
$$v_2 = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} - \frac{1}{2} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1/2 \\ 1 \\ 0 \\ -1/2 \end{pmatrix}$$

Gleiche Methode, nur mehr Komponenten! Funktioniert für jedes n

Was wir gelernt haben:

- Gram-Schmidt macht Vektoren orthonormal
- Funktioniert systematisch Schritt für Schritt
- Anwendbar in \mathbb{R}^n für jedes n

Nächste Frage:

Wie speichern wir diese Transformation als Matrix?

Antwort: QR-Zerlegung

Die QR-Zerlegung ist die “Matrix-Version” von Gram-Schmidt:

$$A = QR$$

wobei Q die orthonormalen Vektoren und R die Koeffizienten speichert.

QR-Zerlegung macht Gram-Schmidt zu einem praktischen Werkzeug

Definition

Jede Matrix $A \in \mathbb{R}^{m \times n}$ mit linear unabhängigen Spalten kann zerlegt werden:

$$A = QR$$

wobei:

- $Q \in \mathbb{R}^{m \times n}$: Spalten sind orthonormal
- $R \in \mathbb{R}^{n \times n}$: obere Dreiecksmatrix

Zusammenhang mit Gram-Schmidt:

- Q enthält die orthonormalen Vektoren q_1, \dots, q_n
- R enthält die Koeffizienten aus der Orthogonalisierung

QR speichert Gram-Schmidt kompakt als Matrixprodukt

Gegeben:

$$A = \begin{pmatrix} 3 & 1 \\ 4 & 2 \end{pmatrix}$$

Spalten: $a_1 = (3, 4)^T$, $a_2 = (1, 2)^T$

Gram-Schmidt liefert (wie vorher berechnet):

$$q_1 = \begin{pmatrix} 0.6 \\ 0.8 \end{pmatrix}, \quad q_2 = \begin{pmatrix} -0.8 \\ 0.6 \end{pmatrix}$$

Also:

$$Q = \begin{pmatrix} 0.6 & -0.8 \\ 0.8 & 0.6 \end{pmatrix}$$

Was ist R? Fortsetzung auf nächster Folie

Q sammelt die orthonormalen Spalten

Wie finden wir R ?

Aus $A = QR$ folgt:

$$R = Q^T A$$

Berechnung:

$$Q^T = \begin{pmatrix} 0.6 & 0.8 \\ -0.8 & 0.6 \end{pmatrix}$$

$$R = Q^T A = \begin{pmatrix} 0.6 & 0.8 \\ -0.8 & 0.6 \end{pmatrix} \begin{pmatrix} 3 & 1 \\ 4 & 2 \end{pmatrix} = \begin{pmatrix} 5 & 2.2 \\ 0 & 0.4 \end{pmatrix}$$

Ergebnis:

$$A = \begin{pmatrix} 3 & 1 \\ 4 & 2 \end{pmatrix} = \begin{pmatrix} 0.6 & -0.8 \\ 0.8 & 0.6 \end{pmatrix} \begin{pmatrix} 5 & 2.2 \\ 0 & 0.4 \end{pmatrix}$$

R ist obere Dreiecksmatrix - das ist immer so!

Warum ist R eine obere Dreiecksmatrix?

Geometrische Erklärung:

Beim Gram-Schmidt-Verfahren gilt:

$$a_i \in \text{span}\{q_1, \dots, q_i\}$$

Das bedeutet:

$$a_i = r_{1i}q_1 + r_{2i}q_2 + \dots + r_{ii}q_i + 0 \cdot q_{i+1} + \dots$$

\Rightarrow Die i -te Spalte von R hat nur Einträge in Zeilen $1, \dots, i$!

Konkret:

$$R = \begin{pmatrix} r_{11} & r_{12} & r_{13} & \cdots \\ 0 & r_{22} & r_{23} & \cdots \\ 0 & 0 & r_{33} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

Obere Dreiecksform kommt von der schrittweisen Konstruktion

Hauptanwendungen:

1. Gleichungssysteme lösen:

$$Ax = b \Rightarrow QRx = b \Rightarrow Rx = Q^T b$$

R ist Dreiecksmatrix \Rightarrow einfach per Rücksubstitution lösbar!

2. Least-Squares-Probleme:

Finde x , das $\|Ax - b\|$ minimiert.

Lösung: $x = R^{-1}Q^T b$ (numerisch stabil!)

3. Orthogonale Basis finden:

Spalten von Q bilden Orthonormalbasis!

QR ist eines der wichtigsten Werkzeuge in numerischer linearer Algebra

Aufgabe

Finden Sie die QR-Zerlegung von:

$$A = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$$

Schritte:

1. Wenden Sie Gram-Schmidt auf die Spalten $(1, 1)^T$ und $(0, 1)^T$ an
2. Bilden Sie Q aus den orthonormalen Vektoren
3. Berechnen Sie $R = Q^T A$
4. Überprüfen Sie: $A = QR$

Tipp: Wir haben diese Vektoren schon in der Gram-Schmidt-Übung behandelt!

Versuchen Sie es selbst

Definition

Eine quadratische Matrix $Q \in \mathbb{R}^{n \times n}$ heißt **orthogonal**, wenn:

$$Q^T Q = I$$

Äquivalente Eigenschaften:

- $Q^{-1} = Q^T$ (Inverse = Transponierte!)
- Spalten sind orthonormal
- Zeilen sind orthonormal
- $\|Qx\| = \|x\|$ für alle x (längenerhaltend)

Beispiel:

$$Q = \begin{pmatrix} 0.6 & -0.8 \\ 0.8 & 0.6 \end{pmatrix} \Rightarrow Q^T Q = I$$

Orthogonale Matrizen erhalten Längen und Winkel

1. Rotation um Winkel θ in \mathbb{R}^2 :

$$R_\theta = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}$$

2. Spiegelung an x-Achse:

$$S = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

3. Permutationsmatrix:

$$P = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Gemeinsam: Alle erhalten Abstände und Winkel!

Orthogonale Matrizen beschreiben starre Bewegungen (Isometrien)

Warum sind orthogonale Matrizen wichtig?

1. Numerische Stabilität:

Konditionszahl: $\kappa(Q) = 1$ (optimal!)

⇒ Keine Verstärkung von Rundungsfehlern

2. Einfache Inverse:

$Q^{-1} = Q^T$ – nur transponieren, nicht aufwendig invertieren!

3. Geometrie bleibt erhalten:

- Abstände: $\|Qx - Qy\| = \|x - y\|$
- Winkel: $(Qx) \cdot (Qy) = x \cdot y$

4. In QR-Zerlegung:

Q aus $A = QR$ ist orthogonal (bei quadratischem A)!

Orthogonale Matrizen kombinieren Eleganz mit numerischer Robustheit

Was wir gelernt haben:

1. Gram-Schmidt-Verfahren:

- Erstellt Orthonormalbasis aus beliebiger Basis
- Schritt für Schritt: Orthogonalisieren + Normieren
- Funktioniert in \mathbb{R}^n für jedes n

2. QR-Zerlegung:

- Matrix-Version von Gram-Schmidt: $A = QR$
- Q : orthonormale Spalten, R : obere Dreiecksmatrix
- Wichtig zum Lösen von Gleichungssystemen

3. Orthogonale Matrizen:

- $Q^T Q = I$, d.h. $Q^{-1} = Q^T$
- Erhalten Längen und Winkel
- Numerisch sehr stabil

Diese Konzepte sind fundamental für viele Bereiche der angewandten Mathematik

Teil I: Skalarprodukte und Orthogonalität

Definition Skalarprodukt

Eine Abbildung $\langle \cdot, \cdot \rangle : V \times V \rightarrow \mathbb{R}$ heißt Skalarprodukt, wenn:

1. Bilinearität:

- $\langle \alpha u + \beta v, w \rangle = \alpha \langle u, w \rangle + \beta \langle v, w \rangle$
- $\langle u, \alpha v + \beta w \rangle = \alpha \langle u, v \rangle + \beta \langle u, w \rangle$

2. Symmetrie: $\langle u, v \rangle = \langle v, u \rangle$

3. Positive Definitheit:

- $\langle v, v \rangle \geq 0$
- $\langle v, v \rangle = 0 \Leftrightarrow v = 0$

Skalarprodukte induzieren Normen und Metriken

Beispiele

1. Standard in \mathbb{R}^n :

$$\langle x, y \rangle = \sum_{i=1}^n x_i y_i$$

2. Gewichtet in \mathbb{R}^n :

$$\langle x, y \rangle_W = x^T W y$$

mit W positiv definit

3. Funktionenraum $C([a, b])$:

$$\langle f, g \rangle = \int_a^b f(x)g(x)dx$$

4. Polynome mit Gewicht:

$$\langle p, q \rangle = \int_{-1}^1 p(x)q(x)w(x)dx$$

Induzierte Norm

$$\|v\| = \sqrt{\langle v, v \rangle}$$

Cauchy-Schwarz-Ungleichung

$$|\langle u, v \rangle| \leq \|u\| \cdot \|v\|$$

Gleichheit genau dann, wenn u, v linear abhängig.

Dreiecksungleichung

$$\|u + v\| \leq \|u\| + \|v\|$$

Die Geometrie des Skalarprodukts verallgemeinert die euklidische Geometrie

Winkel zwischen Vektoren

$$\cos \theta = \frac{\langle u, v \rangle}{\|u\| \cdot \|v\|}$$

für $u, v \neq 0$.

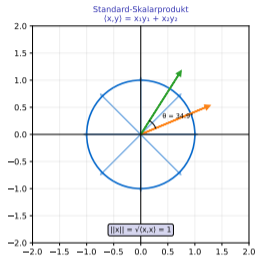
Orthogonalität $u \perp v$ genau dann, wenn $\langle u, v \rangle = 0$

Pythagoras Wenn $u \perp v$:

$$\|u + v\|^2 = \|u\|^2 + \|v\|^2$$

Die Geometrie des Skalarprodukts verallgemeinert die euklidische Geometrie

Skalarprodukte - Geometrie und Beispiele



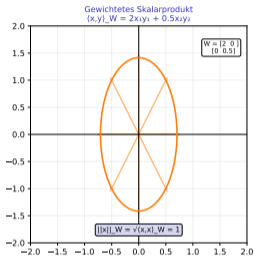
Matrix-Skalarprodukt (Frobenius)

$$\langle A, B \rangle_F = \text{tr}(A^T B) = \sum_i a_{ij} b_{ij}$$

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad B = \begin{bmatrix} 2 & -1 \\ 0 & 3 \end{bmatrix}$$

$$\langle A, B \rangle_F = \text{tr}(A^T B) = 12$$

- Eigenschaften:
- Symmetrisch: $\langle A, B \rangle = \langle B, A \rangle$
 - Linear: $\langle \alpha A + \beta B, C \rangle = \alpha \langle A, C \rangle + \beta \langle B, C \rangle$
 - Positiv definit: $\langle A, A \rangle \geq 0$

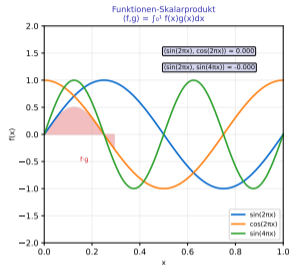


Skalarprodukt-Axiome

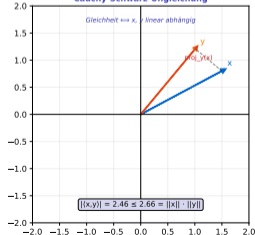
Ein Skalarprodukt $\langle \cdot, \cdot \rangle: V \times V \rightarrow \mathbb{R}$ erfüllt:

1. **Symmetrie:** $\langle x, y \rangle = \langle y, x \rangle$
2. **Linearität:** $\langle \alpha x + \beta y, z \rangle = \alpha \langle x, z \rangle + \beta \langle y, z \rangle$
3. **Positive Definitheit:** $\langle x, x \rangle \geq 0$ und $\langle x, x \rangle = 0 \iff x = 0$

Induzierte Norm: $\|x\| = \sqrt{\langle x, x \rangle}$



Cauchy-Schwarz-Ungleichung



Teil II: Gram-Schmidt-Verfahren

Ziel

Aus linear unabhängigen Vektoren $\{a_1, \dots, a_n\}$ eine Orthonormalbasis $\{q_1, \dots, q_n\}$ konstruieren.

Algorithmus

1. $v_1 = a_1$
2. Für $i = 2, \dots, n$:

$$v_i = a_i - \sum_{j=1}^{i-1} \frac{\langle a_i, v_j \rangle}{\langle v_j, v_j \rangle} v_j$$

3. Normierung:

$$q_i = \frac{v_i}{\|v_i\|}$$

Gram-Schmidt orthogonalisiert schrittweise durch Subtraktion von Projektionen

Geometrische Interpretation

Jeder Schritt:

- Subtrahiere Projektionen auf bereits orthogonalisierte Vektoren
- Ergebnis steht senkrecht auf allen vorherigen

Eigenschaften

- $\text{span}\{a_1, \dots, a_k\} = \text{span}\{q_1, \dots, q_k\}$
- Erhält die Reihenfolge der Vektoren
- Numerisch sensitiv (siehe modifiziertes GS)

Gram-Schmidt orthogonalisiert schrittweise durch Subtraktion von Projektionen

Gegeben

$$a_1 = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}, a_2 = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}, a_3 = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$$

Schritt 1: $v_1 = a_1 = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$

Schritt 2:

$$v_2 = a_2 - \frac{\langle a_2, v_1 \rangle}{\langle v_1, v_1 \rangle} v_1 \tag{1}$$

$$= \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} - \frac{1}{2} \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \tag{2}$$

$$= \begin{pmatrix} 1/2 \\ -1/2 \\ 1 \end{pmatrix} \tag{3}$$

Jeder neue Vektor wird orthogonal zu allen vorherigen gemacht

Schritt 3:

$$v_3 = a_3 - \text{proj}_{v_1}(a_3) - \text{proj}_{v_2}(a_3) \quad (4)$$

$$= \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} - \frac{1}{2} \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} - \frac{1}{3} \begin{pmatrix} 1/2 \\ -1/2 \\ 1 \end{pmatrix} \quad (5)$$

$$= \begin{pmatrix} -2/3 \\ 2/3 \\ 2/3 \end{pmatrix} \quad (6)$$

Normierung:

$$q_1 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}, q_2 = \frac{1}{\sqrt{6}} \begin{pmatrix} 1 \\ -1 \\ 2 \end{pmatrix}, q_3 = \frac{1}{\sqrt{3}} \begin{pmatrix} -1 \\ 1 \\ 1 \end{pmatrix}$$

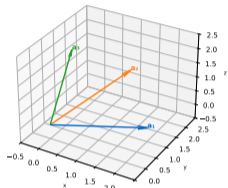
Jeder neue Vektor wird orthogonal zu allen vorherigen gemacht

Gram-Schmidt: Schrittweise Orthogonalisierung im 3D

Gram-Schmidt-Verfahren: Schritt-für-Schritt Orthogonalisierung

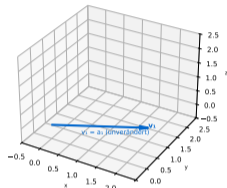
- Vektor 1
- Vektor 2
- Vektor 3
- Projektionen

Schritt 0: Ausgangsvektoren



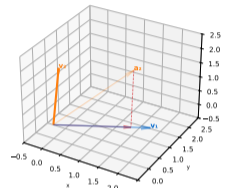
Schritt 3: $v_3 = a_3 - \text{proj}_{\{v_1\}}\{a_3\} - \text{proj}_{\{v_2\}}\{a_3\}$

Schritt 1: $v_1 = a_1$

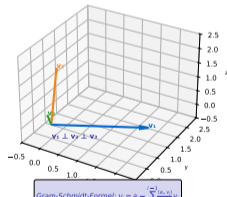
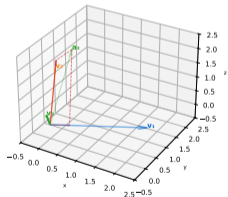


Schritt 4: Orthogonale Vektoren

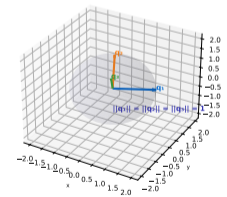
Schritt 2: $v_2 = a_2 - \text{proj}_{\{v_1\}}\{a_2\}$



Schritt 5: Orthonormale Vektoren



Gram-Schmidt-Formel: $v_i = a_i - \sum_{j=1}^{i-1} \frac{\langle a_i, v_j \rangle}{\langle v_j, v_j \rangle} v_j$
 Normierung: $q_i = \frac{v_i}{\|v_i\|}$



Schrittweise Orthogonalisierung im 3D-Raum

Klassisches Gram-Schmidt (CGS) “ for $i = 1$ to n : $v[i] = a[i]$ for $j = 1$ to $i-1$: $v[i] = v[i] - \frac{a[i] \cdot v[j]}{v[j] \cdot v[j]} * v[j]$ $q[i] = v[i] / \|v[i]\|$ “

Problem: Rundungsfehler akkumulieren

Verlust der Orthogonalität

Bei fast linear abhängigen Vektoren:

- Kleine Differenzen werden ungenau
- Orthogonalität geht verloren

Modifiziertes Gram-Schmidt ist numerisch stabiler als die klassische Variante

Modifiziertes Gram-Schmidt (MGS) ““ for $i = 1$ to n : $v[i] = a[i]$ for $i = 1$ to n : $q[i] = v[i] / \|v[i]\|$ for $j = i+1$ to n : $v[j] = v[j] - \langle v[j], q[i] \rangle q[i]$ ““

Vorteile

- Bessere numerische Stabilität
- Gleiche Komplexität $O(mn^2)$
- Orthogonalität bleibt besser erhalten

Alternative: Householder-QR (noch stabiler)

Modifiziertes Gram-Schmidt ist numerisch stabiler als die klassische Variante

Satz (QR-Zerlegung)

Sei $A \in \mathbb{R}^{m \times n}$ mit linear unabhängigen Spalten. Dann existiert eine eindeutige Zerlegung:

$$A = QR$$

mit:

- $Q \in \mathbb{R}^{m \times n}$: orthonormale Spalten
- $R \in \mathbb{R}^{n \times n}$: obere Dreiecksmatrix mit $r_{ii} > 0$

Zusammenhang mit Gram-Schmidt

Q enthält die orthonormalisierten Spalten R enthält die Koeffizienten der Darstellung

Die QR-Zerlegung faktorisiert eine Matrix in orthogonale und obere Dreiecksmatrix

Berechnung von R

Aus $A = QR$ folgt $Q^T A = R$:

$$r_{ij} = \begin{cases} \langle a_j, q_i \rangle & i \leq j \\ 0 & i > j \end{cases}$$

Eigenschaften

- $Q^T Q = I_n$ (orthonormale Spalten)
- $\det(A) = \det(R) = \prod_{i=1}^n r_{ii}$
- Wenn $m = n$: Q orthogonal, $Q^{-1} = Q^T$

Die QR-Zerlegung faktorisiert eine Matrix in orthogonale und obere Dreiecksmatrix

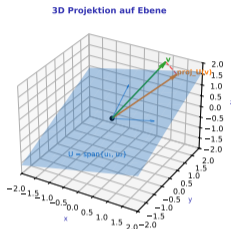
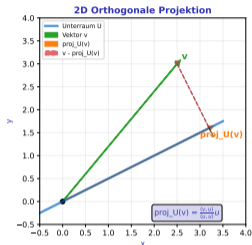
QR-Zerlegung: Struktur und Eigenschaften

$$\text{Fundamentale Gleichung: } A = QR$$



Struktur und Eigenschaften der QR-Zerlegung

Orthogonale Projektion



Projektionsmatrix

$$P = QQ^T \quad (Q \text{ hat orthonormale Spalten})$$

Eigenschaften:

- Idempotent: $P^2 = P$
- Symmetrisch: $P^T = P$
- Eigenwerte: 0 und 1
- $\text{rang}(P) = \dim(U)$

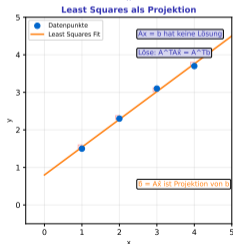
Berechnung:

$\text{proj}_U(v) = Pv$
 Für $U = \text{span}\{u\}$:
 $P = \frac{uu^T}{u^Tu}$
 Orthogonal: $v - Pv$

Orthogonale Zerlegung:

$$v = Pv + (I-P)v$$

$\in U \quad \in U^\perp$



Gram-Schmidt als Projektion

$$v_1 = a_1$$

$$v_2 = a_2 - \text{proj}_{\{v_1\}}(a_2)$$

$$v_3 = a_3 - \text{proj}_{\{v_1\}}(a_3) - \text{proj}_{\{v_2\}}(a_3)$$

$$v_i = a_i - \sum_{j=1}^{i-1} \langle a_i, v_j \rangle v_j$$

Jeder Schritt projiziert orthogonal zum bisherigen Raum

Anwendungen

- Least-Squares-Approximation
- Signalverarbeitung (Fourier)
- Computer Graphics (3D \rightarrow 2D)
- Statistik (PCA)
- Maschinelles Lernen
- Numerische Lineare Algebra

$$\text{Allgemein: } P = Q(Q^T Q)^{-1} Q^T$$

$(Q \text{ Spalten} = \text{Basis von } U)$

Projektion auf Unterräume und Least-Squares

Teil III: Orthogonale Matrizen und Abbildungen

Definition

$Q \in \mathbb{R}^{n \times n}$ heißt orthogonal, wenn:

$$Q^T Q = Q Q^T = I$$

Äquivalent:

- $Q^{-1} = Q^T$
- Spalten bilden ONB
- Zeilen bilden ONB
- $\langle Qx, Qy \rangle = \langle x, y \rangle$

Eigenschaften

- $\det(Q) = \pm 1$
- Eigenwerte: $|\lambda| = 1$
- Erhält Längen: $\|Qx\| = \|x\|$
- Erhält Winkel

Orthogonale Matrizen beschreiben längenerhaltende lineare Abbildungen

Fortgeschrittene Beispiele

Householder-Matrix ($v = (1, 0)^T$):

$$H = I - 2vv^T = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$$

3D-Rotation um z-Achse ($\theta = 90$):

$$R_z = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

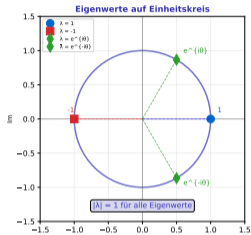
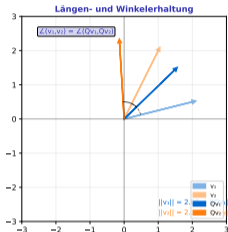
Blockdiagonale orthogonale Matrix:

$$Q = \begin{pmatrix} R_{2 \times 2} & 0 \\ 0 & R_{2 \times 2} \end{pmatrix}$$

Gruppenstruktur Orthogonale Matrizen bilden die Gruppe $O(n)$ mit $\det(Q) = \pm 1$

Orthogonale Matrizen beschreiben längenerhaltende lineare Abbildungen

Orthogonale Matrizen - Eigenschaften und Geometrie



Eigenschaften orthogonaler Matrizen

Definition:
• $Q^{-T}Q = QQ^T = \theta + (-1) = Q^{-T}$

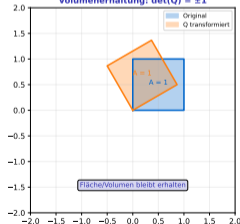
Spalten/Zeilen:
• Orthonormale Spalten/Zeilen

Determinante:
• $\det(Q) = \pm 1$
• $+1$: Rotation
• -1 : Spiegelung

Eigenwerte:
• $|\lambda| = 1$
• λ reell: ± 1
• λ komplex: konjugiert

Erhaltung:
• $\|Qx\| = \|x\|$
• $\langle Qx, Qy \rangle = \langle x, y \rangle$
• $L(Qx, Qy) = L(x, y)$

Volumenerhaltung: $\det(Q) = \pm 1$



Spezielle orthogonale Matrizen

2D Rotation:

$$R(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$$

2D Spiegelung:

$$S = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

Householder:

$$H = I - 2 \frac{vv^T}{v^T v}$$

Spiegelung an Hyperebene

Givens-Rotation:

Rotation in 2D-Unterraum
Verwendet für QR-Zerlegung

Permutationsmatrix:

Vertauscht Zeilen/Spalten
Genau eine 1 pro Zeile/Spalte



Rotation

Verbindung zur QR-Zerlegung

$$A = QR$$

Q orthogonal, R obere Dreiecksmatrix

Gram-Schmidt:

1. Orthogonalisiere Spalten von A : $Q^T A = R$
2. Normiere zu Einheitsvektoren: Numerisch stabil
3. Diese bilden Q

Eigenschaften:

- Erhält Kondition

Anwendungen in der Numerik:

- Lösen linearer Gleichungssysteme
- Eigenwertberechnung
- Squares-Probleme
- Konditionsverbesserung

Geometrische und algebraische Eigenschaften orthogonaler Transformationen

Definition

Householder-Matrix (Spiegelung):

$$H = I - 2vv^T$$

mit $\|v\| = 1$.

Eigenschaften

- $H^T = H$ (symmetrisch)
- $H^2 = I$ (involutorisch)
- $\det(H) = -1$ (Spiegelung)
- $Hv = -v$ (v wird gespiegelt)
- $Hw = w$ für $w \perp v$

Householder-Transformationen spiegeln Vektoren auf Koordinatenachsen

Anwendung: QR-Zerlegung

Ziel: Spalte auf e_1 -Richtung abbilden

Für $x \neq 0$, wähle:

$$v = \frac{x - \|x\|e_1}{\|x - \|x\|e_1\|}$$

Dann: $Hx = \|x\|e_1$

Algorithmus

1. Transformiere erste Spalte
2. Wende auf Restmatrix an
3. Iteriere für kleinere Matrix

Komplexität: $O(2n^3/3)$

Householder-Transformationen spiegeln Vektoren auf Koordinatenachsen

Definition

Givens-Matrix (Rotation in Ebene i, j):

$$G_{ij} = \begin{pmatrix} I & & & & \\ & c & \cdots & & -s \\ & \vdots & I & & \vdots \\ & s & \cdots & & c \\ & & & & & I \end{pmatrix}$$

mit $c^2 + s^2 = 1$.

Position (i, i) : c , Position (j, j) : c Position (i, j) : $-s$, Position (j, i) : s

Givens-Rotationen nullen gezielt einzelne Matrixelemente

Anwendung

Nullsetzen von Element (j, i) in Vektor:

$$\begin{pmatrix} c & -s \\ s & c \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} r \\ 0 \end{pmatrix}$$

mit:

$$c = \frac{a}{\sqrt{a^2 + b^2}} \quad (7)$$

$$s = \frac{b}{\sqrt{a^2 + b^2}} \quad (8)$$

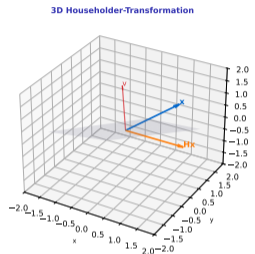
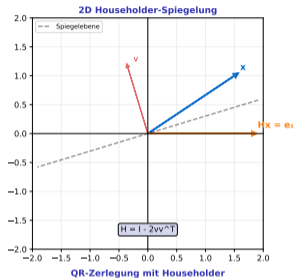
$$r = \sqrt{a^2 + b^2} \quad (9)$$

Vorteil: Spärlich, lokal **Nachteil:** Mehr Operationen als Householder

Givens-Rotationen nullen gezielt einzelne Matrixelemente

Householder-Spiegelung: Geometrische Interpretation

Householder-Transformationen für QR-Zerlegung



Vorteile der Householder-Methode

Vergleich der QR-Methoden

Methode	Stabilität	Geschwindigkeit	Speicher
Gram-Schmidt	Schlecht	$O(2mn^2)$	Q und R
Mod. Gram-Schmidt	Besser	$O(2mn^2)$	Q und R
Householder	Exzellente	$O(2mn^2 - 2n^2/3)$	Nur R
Givens	Exzellente	$O(3mn^2 - n^2)$	Nur R

Householder-Vorteile:

- ✓ Numerisch stabil (orthogonale Transformationen)
- ✓ Benötigt nur R zu speichern (Q implizit)
- ✓ Gut für dichte Matrizen
- ✓ Parallelisierbar

Householder-Matrix Struktur

$$H = I - 2vv^T / \|v\|^2$$

- Symmetrisch: $H^T = H$
- Orthogonal: $H^T H = I$
- Involutorisch: $H^2 = I$
- $\det(H) = -1$ (Spiegelung)

$$H = \begin{bmatrix} 1-2v_1^2 & 2v_1v_2 & 2v_1v_3 \\ 2v_2v_1 & 1-2v_2^2 & 2v_2v_3 \\ 2v_3v_1 & 2v_3v_2 & 1-2v_3^2 \end{bmatrix}$$

Kosten: $O(n^3)$

Beispielrechnung

Gegeben: $x = [3, 4]^T$

$$\|x\| = 5$$

Ziel: $Hx = [5, 0]^T$

$$v = x - \|x\|e_1 = [3, 4]^T - [5, 0]^T = [-2, 4]^T$$

$$v = v/\|v\| = [-2, 4]^T \cdot \sqrt{20}$$

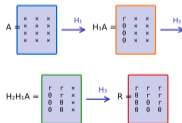
$$H = I - 2vv^T = \begin{bmatrix} 0.6 & 0.8 \\ -0.8 & 0.6 \end{bmatrix}$$

$$Hx = \begin{bmatrix} 0.6 & 0.8 \\ -0.8 & 0.6 \end{bmatrix} \begin{bmatrix} 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 5 \\ 0 \end{bmatrix}$$

$$[-0.8 \ 0.6] \begin{bmatrix} 3 \\ 4 \end{bmatrix} = [0]$$

✓ Vektor auf e_1 -Richtung gespiegelt

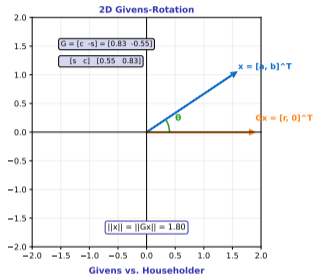
Schrittweise Transformation



$$Q^T = H_{1,n} \dots H_2 H_1 \rightarrow A = QR$$

Vergleich der beiden Hauptmethoden für QR-Zerlegung

Givens-Rotationen für QR-Zerlegung



	Givens	Householder
Strategie	Lokale Rotationen	Globale Spiegelungen
Nullen pro Schritt	1 Element	Ganze Spalte
Anzahl Schritte	$n(n-1)/2$	n
Speicherbedarf	Minimal	Vektor pro Spalte
Parallelisierung	Gut	Begrenzt
Dünnbesetzte Matrizen	Optimal	Suboptimal

Anwendungsempfehlung:

- Givens: Dünnbesetzte Matrizen
- Householder: Dichte Matrizen

Givens-Matrix Struktur (i,j)-Rotation



Beispiel: Elimination von a_{21}

Gegeben: $a_{11} = 3, a_{21} = 4$

$$r = \sqrt{a_{11}^2 + a_{21}^2} = \sqrt{9 + 16} = 5$$

$$c = a_{11}/r = 3/5 = 0.6$$

$$s = a_{21}/r = 4/5 = 0.8$$

$$G = [0.6 \ -0.8]$$

$$[0.8 \ 0.6]$$

Resultat:

$$G \cdot \begin{bmatrix} 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 5 \\ 0 \end{bmatrix}$$

✓ Element a_{21} erfolgreich eliminiert

QR-Zerlegung mit Givens-Rotationen

Sukzessive Elimination



Eliminationsreihenfolge:

$$(2,1) = (3,1) = (4,1) = \dots$$

$$(3,2) = (4,2) = \dots$$

$$(4,3) = \dots$$

Anzahl Rotationen: $n(n-1)/2$

Anwendung: Dünnbesetzte Matrizen

Tridiagonalmatrix



- Givens erhält Dünnbesetztheit
- Nur $O(n)$ Operationen nötig
- Householder würde Füll-in erzeugen
- Ideal für Band- und Hessenberg-Matrizen

Givens optimal für strukturierte Matrizen

Problem Löse $Ax = b$ mit $A \in \mathbb{R}^{m \times n}$, $m \geq n$

QR-Methode

1. Berechne $A = QR$
2. Multipliziere mit Q^T :

$$Q^T Ax = Q^T b$$

$$Rx = Q^T b$$

3. Löse durch Rücksubstitution

Vorteile

- Numerisch stabil
- Funktioniert für $m > n$ (Least Squares)
- Kondition: $\kappa(R) = \kappa(A)$

QR-Zerlegung ermöglicht stabiles Lösen linearer Systeme

Least-Squares-Problem

Minimiere $\|Ax - b\|^2$ für überbestimmtes System

Lösung via QR:

$$\|Ax - b\|^2 = \|QRx - b\|^2 = \|Rx - Q^T b\|^2$$

Zerlege $Q^T b = \begin{pmatrix} c \\ d \end{pmatrix}$ mit $c \in \mathbb{R}^n$

Dann: $\|Ax - b\|^2 = \|Rx - c\|^2 + \|d\|^2$

Minimum bei $Rx = c$

Normalgleichungen Alternative: $A^T Ax = A^T b$ Aber: Kondition schlechter!

QR-Zerlegung ermöglicht stabiles Lösen linearer Systeme

Lösen linearer Gleichungssysteme mit QR-Zerlegung

Problemstellung: $Ax = b$

Gegeben: Überbestimmtes System

$$A = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 2 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 2 \end{bmatrix} \quad x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad b = \begin{bmatrix} 2 \\ 3 \\ 1 \\ 4 \end{bmatrix}$$

$m > n$: Mehr Gleichungen als Unbekannte

Lösung: Minimiere $\|Ax - b\|_2$

Schritt 3: Rücksubstitution

Oberes Dreieckssystem: $Rx = c$

$$R = \begin{bmatrix} 2 & 1 & 3 \\ 0 & 3 & 2 \\ 0 & 0 & 4 \end{bmatrix} \quad x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad c = \begin{bmatrix} 5 \\ 7 \\ 8 \end{bmatrix}$$

Rücksubstitution (von unten nach oben):

- $x_3 = c_3/r_{33} = 8/4 = 2$
- $x_2 = (c_2 - r_{23}x_3)/r_{22} = (7 - 2 \cdot 2)/3 = 1$
- $x_1 = (c_1 - r_{12}x_2 - r_{13}x_3)/r_{11} = (5 - 1 \cdot 1 - 3 \cdot 2)/2 = -1$

Lösung: $x = [-1, 1, 2]^T$

Schritt 1: QR-Zerlegung von A

$A = QR$

$$A = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 2 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 2 \end{bmatrix} = Q = \begin{bmatrix} q_1 & q_2 & q_3 \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \end{bmatrix} \times R = \begin{bmatrix} r & r & r \\ 0 & r & r \\ 0 & 0 & r \end{bmatrix}$$

Q: $m \times n$, orthonormale Spalten
R: $n \times n$, obere Dreiecksmatrix

$Ax = b \rightarrow QRx = b \rightarrow Rx = Q^T b$

QR-Lösungsalgorithmus

Algorithmus: Löse $Ax = b$ mit QR

- QR-Zerlegung:** $A = QR$
- Transformation:** $c = Q^T b$
- Dreieckssystem:** $Rx = c$
- Rücksubstitution:** x lösen

Komplexität: $O(2mn^2 - 2n^3/3)$

Numerisch stabil durch orthogonale Transformationen

Schritt 2: Transformation zu R

Original:

$QRx = b$

$Q^T Q^T x = c$

Multipliziere mit Q^{-T}
 3×4 4×1 3×1

Transformiert:

$Rx = Q^T b$

$Rx = c$

Beachte: $Q^T Q = I$ (orthonormal)
 \Rightarrow Erhält Länge: $\|c\| = \|Q^T b\| = \|b\|$

Vollständiges Beispiel

Löse: $Ax = b$

$$A = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad b = \begin{bmatrix} 3 \\ 1 \\ 2 \end{bmatrix}$$

1. QR-Zerlegung:

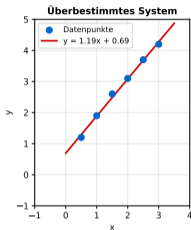
$$Q = \begin{bmatrix} -0.71 & -0.41 \\ -0.71 & 0.82 \end{bmatrix}$$

2. $Q^T b = c$:

$$c = \begin{bmatrix} -2.83 \\ 0.82 \end{bmatrix}$$

3. Löse $Rx = c$:
 $x = [1, 2]^T$

Verifikation: $\|Ax - b\|$ minimal



QR-Lösung

QR-Zerlegung: $A = QR$

1. Berechne QR-Zerlegung
2. Löse: $Rx = Q^T b$
(Rücksubstitution)

Vorteile:

- Numerisch stabiler
- Vermeidet $A^T A$
- Konditionszahl $\sqrt{\kappa(A^T A)} \rightarrow \kappa(A)$

Least Squares: Überbestimmte Systeme optimal lösen

Normalgleichungen

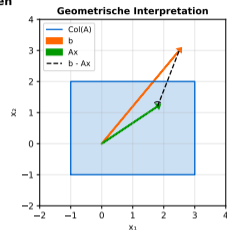
$$A^T A x = A^T b$$

Lösung:

$$x = (A^T A)^{-1} A^T b$$

Eigenschaften:

- Minimiert $\|Ax - b\|^2$
- Eindeutig wenn A vollen Rang hat
- $A^T A$ symmetrisch positiv definit



Fehleranalyse

Residuum:

$$r = b - Ax_{LS}$$

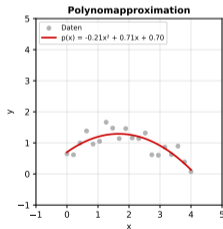
Minimierung:

$$\|r\|_2^2 = \|b - Ax_{LS}\|_2^2$$

Orthogonalität:

$$A^T r = 0$$

(Residuum \perp Col(A))



Geometrische Interpretation und numerische Verfahren für überbestimmte Systeme

Teil IV: Diskrete Cosinus-Transformation (DCT)

Motivation

Signalkompression durch:

- Transformation in Frequenzbereich
- Energiekompaktierung
- Weglassen kleiner Koeffizienten

Warum DCT statt DFT?

- Nur reelle Werte
- Bessere Energiekompaktierung
- Keine Randeffekte (gerade Fortsetzung)
- Schnelle Algorithmen verfügbar

DCT ist die Basis vieler moderner Kompressionsverfahren

Anwendungen

- **JPEG**: Bildkompression
- **MP3**: Audiokompression (MDCT)
- **Video**: H.264, MPEG
- Spracherkennung
- Mustererkennung

JPEG-Pipeline

1. Farbraumkonversion (RGB \rightarrow YCbCr)
2. 8×8 Block-DCT
3. Quantisierung
4. Entropiekodierung

DCT ist die Basis vieler moderner Kompressionsverfahren

DCT-II (Standard)

Für N Punkte, Basisfunktionen:

$$\phi_k[n] = \alpha_k \cos\left(\frac{\pi k(2n+1)}{2N}\right)$$

mit $n = 0, \dots, N-1$ und

$$\alpha_k = \begin{cases} \sqrt{1/N} & k = 0 \\ \sqrt{2/N} & k > 0 \end{cases}$$

Eigenschaften

- Orthonormal: $\langle \phi_i, \phi_j \rangle = \delta_{ij}$
- Reellwertig
- Gerade Symmetrie

DCT-Basisfunktionen bilden eine orthonormale Basis aus Cosinusfunktionen

Matrix-Darstellung

DCT-Matrix $C \in \mathbb{R}^{N \times N}$:

$$C_{kn} = \alpha_k \cos\left(\frac{\pi k(2n+1)}{2N}\right)$$

Transformation

- Vorwärts: $X = CX$
- Rückwärts: $x = C^T X$ (da $C^T C = I$)

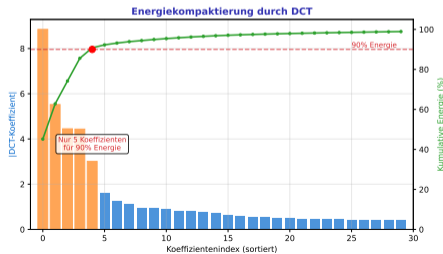
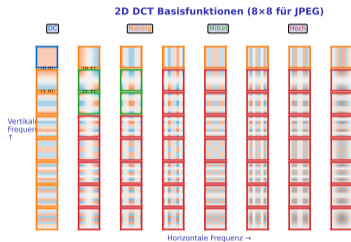
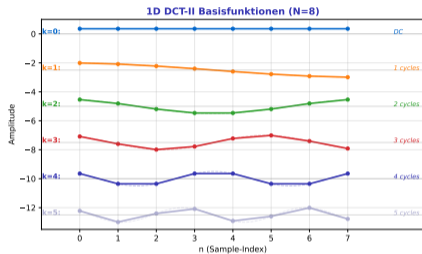
2D-DCT (für Bilder)

$$X = CXC^T$$

Separabel: erst Zeilen, dann Spalten

DCT-Basisfunktionen bilden eine orthonormale Basis aus Cosinusfunktionen

Diskrete Cosinus-Transformation (DCT) - Basisfunktionen



DCT-II Transformation

1D-DCT:

$$X_k = a_k \sum_{n=0}^{N-1} x_n \cos\left(\frac{(n+1/2)k\pi}{N}\right)$$

$$a_k = \sqrt{1/N} \text{ für } k=0, \quad a_k = \sqrt{2/N} \text{ für } k>0$$

2D-DCT (JPEG):

$$X_{k,l} = a_k a_l \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x_{m,n} \cos\left(\frac{(m+1/2)k\pi}{M}\right) \cos\left(\frac{(n+1/2)l\pi}{N}\right)$$

Eigenschaften

- Orthonormale Basis
- Energiekompaktierung
- Fast-DCT: $O(N \log N)$
- Reellwertig
- Separabel (2D = 1D × 1D)
- Basis für JPEG

JPEG: 8×8 Blöcke

Bild wird in 8×8 Pixel-Blöcke zerlegt

Ablauf pro Block:

1. Verschiebung: $[0, 255] \rightarrow [-128, 127]$
2. 2D-DCT anwenden
3. Quantisierung mit Matrix Q :

$$X_q = \text{round}(X/Q)$$

4. Zig-Zag-Scan für 1D-Folge
5. Entropiekodierung

Energiekompaktierung

Meiste Energie in oberen linken Koeffizienten (niedrige Frequenzen)

JPEG nutzt 8×8 Block-DCT mit anschließender Quantisierung

Quantisierungsmatrix

Beispiel (Luminanz, Qualität 50):

$$Q = \begin{pmatrix} 16 & 11 & 10 & 16 & \dots \\ 12 & 12 & 14 & 19 & \dots \\ 14 & 13 & 16 & 24 & \dots \\ 14 & 17 & 22 & 29 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

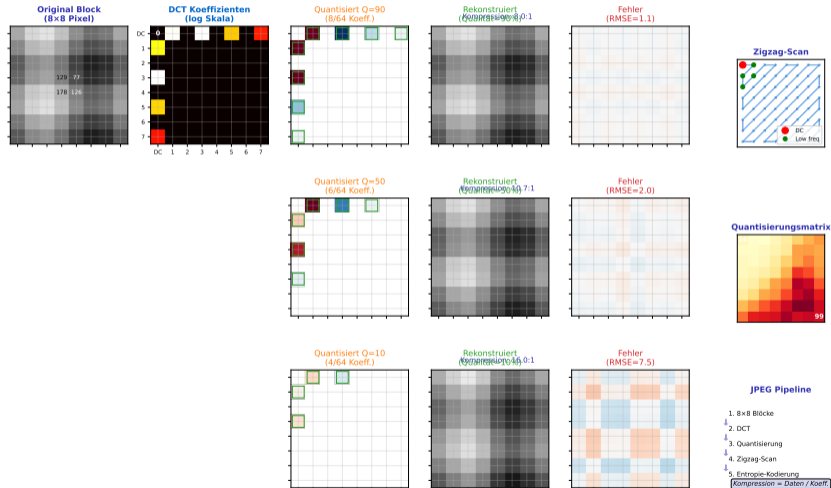
Höhere Werte \rightarrow mehr Kompression

Qualität vs. Dateigröße

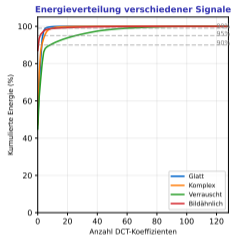
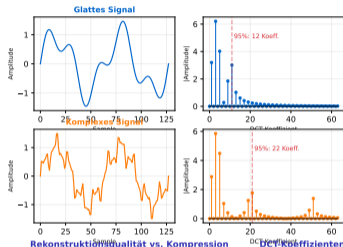
- Qualität 100: kaum Verlust, große Datei
- Qualität 75: guter Kompromiss
- Qualität 25: sichtbare Artefakte

JPEG nutzt 8×8 Block-DCT mit anschließender Quantisierung

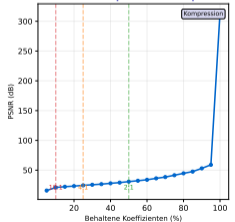
JPEG-Kompression mit DCT - 8×8 Block Demo



DCT Energiekompaktierung - Grundlage der Kompression



Rekonstruktionsqualität vs. Kompression



DCT Koeffizientenwichtigkeit (8x8 Block)



Prinzip der Energiekompaktierung

DCT konzentriert Signalenergie
 Wenige große Koeffizienten → Hohe Kompression



1D-Beispiel

Signal: $x = [4, 3, 5, 10, 5, 3, 2, 1]$

DCT-Koeffizienten:

$$X = [13.4, -3.2, 2.3, -5.7, 0.3, -0.1, 0.8, -0.5]$$

Beobachtung:

- DC-Komponente (Mittelwert) dominiert
- Höhere Frequenzen klein
- Erste 3-4 Koeffizienten genügen oft

DCT konzentriert die Signalenergie in wenigen Koeffizienten

Kompression durch Nullsetzen

Behalte nur größte k Koeffizienten:

- $k = 8$: perfekte Rekonstruktion
- $k = 4$: $\approx 90\%$ der Energie
- $k = 2$: grobe Approximation

Mathematische Eigenschaften

- Orthogonale Transformation
- Energieerhaltend
- Dekorrelation benachbarter Pixel
- Fast-DCT: $O(N \log N)$

DCT konzentriert die Signalenergie in wenigen Koeffizienten

Orthogonalität

- Skalarprodukte definieren Geometrie
- Orthonormalbasen vereinfachen Berechnungen

Gram-Schmidt und QR

- Systematische Orthogonalisierung
- QR-Zerlegung: $A = QR$
- Numerisch stabile Algorithmen wichtig

Orthogonalität ist ein zentrales Konzept mit vielfältigen Anwendungen

Orthogonale Matrizen

- Längen- und winkelerhaltend
- Householder und Givens für QR
- Effizientes Lösen von Gleichungssystemen

DCT und Anwendungen

- Orthogonale Transformation mit Cosinus-Basis
- Energiekompaktierung für Kompression
- Basis von JPEG und vielen anderen Standards

Von Gram-Schmidt bis JPEG: Orthogonalität in Theorie und Praxis

Aufgabe 1: Gram-Schmidt

Orthogonalisieren Sie:

$$a_1 = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}, a_2 = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}, a_3 = \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix}$$

Aufgabe 2: QR-Zerlegung

Bestimmen Sie Q und R für:

$$A = \begin{pmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Üben Sie die verschiedenen Orthogonalisierungsverfahren und ihre Anwendungen

Aufgabe 3: Orthogonale Matrix

Zeigen Sie, dass für Q orthogonal gilt:

$$\|Qx - Qy\| = \|x - y\|$$

Aufgabe 4: DCT

Berechnen Sie die 4-Punkt DCT von:

$$x = [8, 4, 2, 2]$$

Welche Koeffizienten können für Kompression weggelassen werden?

Diese Aufgaben vertiefen das Verständnis für orthogonale Transformationen