

Data Mining

Dimensionsreduktion

Dr. Hanna Köpcke
Wintersemester 2020

Abteilung Datenbanken, Universität Leipzig
<http://dbs.uni-leipzig.de>

Übersicht

Hochdimensionale Daten

Clustering

Dimensions-
reduktion

Empfehlungs-
systeme

Assoziations-
regeln

Locality Sensitive
Hashing

Supervised ML

Graphdaten

Community
Detection

PageRank

Web Spam

Datenströme

Windowing

Filtern

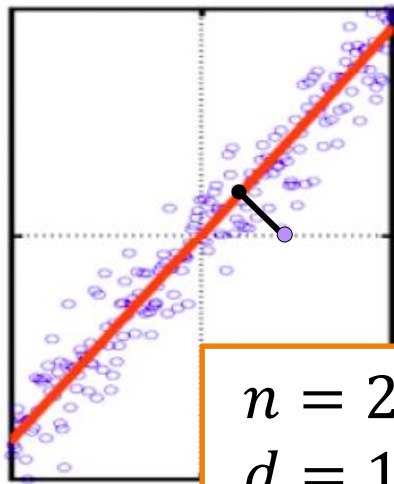
Momente

Web Advertising

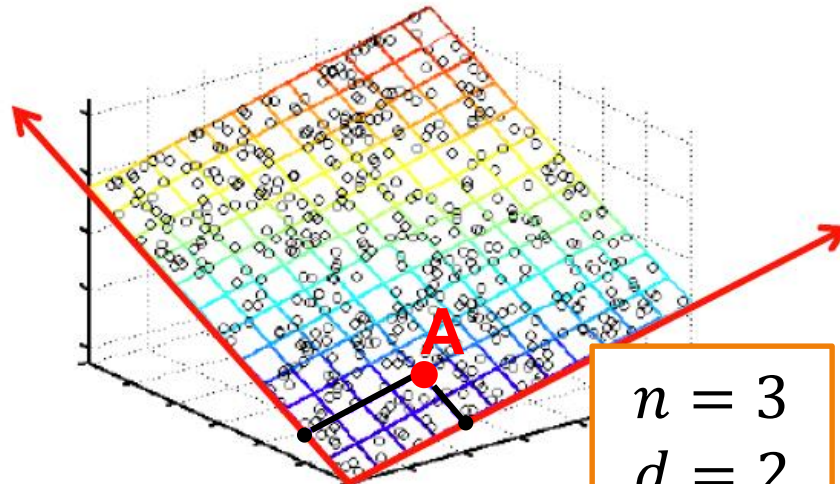
Inhaltsverzeichnis

- **Einführung**
- **Hauptkomponentenanalyse**
- **Singulärwertzerlegung**
- **CUR-Zerlegung**

Dimensionsreduktion



$$\begin{aligned} n &= 2 \\ d &= 1 \end{aligned}$$



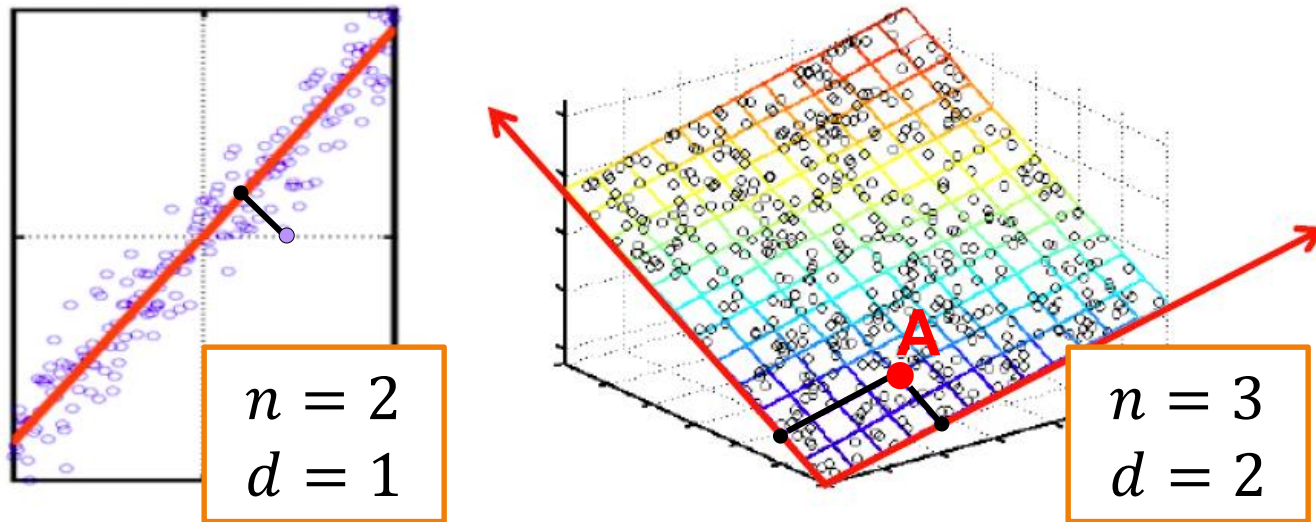
$$\begin{aligned} n &= 3 \\ d &= 2 \end{aligned}$$

- **Idee:** Falls die Datenpunkte eines n -dimensionalen Raumes in der Nähe eines d -dimensionalen *Unterraums* liegen, Reduzierung der Punkte auf deren Projektionen im Unterraum

Anstatt 2 Koordinaten, wird jeder Punkt nur über eine Koordinate repräsentieren: Position auf der roten Linie

Punkt A wird anstatt über 3 Koordinaten (z.B. [3,4,2]) nur über 2 Koordinaten (z.B. [2,1]) repräsentiert

Dimensionsreduktion



Ziel: Aufdeckung der Datenachsen

- Die Achsen des Unterraums bezeichnet man auch als **Faktoren**
- Wahl der Faktoren:
 - Erster Faktor zeigt in die Richtung, in welcher die Punkte ihre größte Streuung aufweisen
 - Zweiter Faktor ist orthogonal zum ersten Faktor und zeigt in die Richtung mit der größten Streuung unter den Punkten
 - USW..

Dimensionsreduktion

- Einfaches Beispiel:

Kunde	Montag	Dienstag	Mittwoch	Donnerstag	Freitag	Repräsentation
A	1	1	1	0	0	[1, 0]
B	2	2	2	0	0	[2, 0]
C	1	1	1	0	0	[1, 0]
D	5	5	5	0	0	[5, 0]
E	0	0	0	2	2	[0, 2]
F	0	0	0	3	3	[0, 3]

- Rang** einer Matrix: Anzahl der *linear unabhängigen* Zeilen/Spalten
 - Im Beispiel: Zeilen-/Spaltenrang ist 2

– Aufspaltung:

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 2 & 0 \\ 1 & 0 \\ 5 & 0 \\ 0 & 2 \\ 0 & 3 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

Neue Achsen/Faktoren

Neue Koordinaten

Inhaltsverzeichnis

- Einführung
- **Hauptkomponentenanalyse**
- Singulärwertzerlegung
- CUR-Zerlegung

Hauptkomponentenanalyse

- Principal Component Analysis (**PCA**)
- Sei A eine Matrix mit m Zeilen (Datenpunkte) und n Spalten (Dimensionen) und $A_{.1}, A_{.2}, \dots, A_{.n}$ die Spalten von A
- *Annahme*: Spalten sind zentriert, so dass

$$\frac{1}{m} \sum_{i=1}^m A_{ij} = 0 \text{ für alle } j = 1, \dots, n$$

- *Erste Hauptkomponente* von A ist gegeben durch den Vektor

$$Z_{.1} = \varphi_{11} A_{.1} + \varphi_{21} A_{.2} + \dots + \varphi_{n1} A_{.n}$$

mit $\sum_{j=1}^n \varphi_{j1}^2 = 1$ und maximaler empirischer Varianz

$$\frac{1}{m} \sum_{i=1}^m Z_{i1}^2$$

- Die Parameter $\varphi_{11}, \varphi_{21}, \dots, \varphi_{n1}$ werden als *Ladungen* der ersten Hauptkomponente bezeichnet

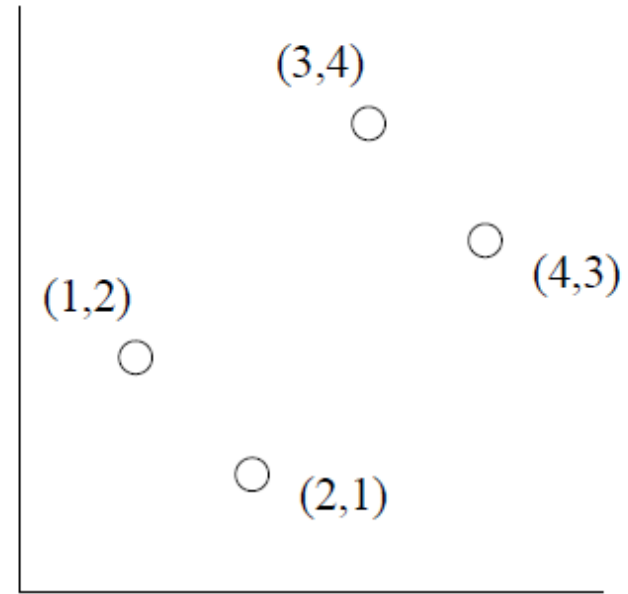
PCA: Beispiel

- 4 Beobachtungen und 2 Dimensionen

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

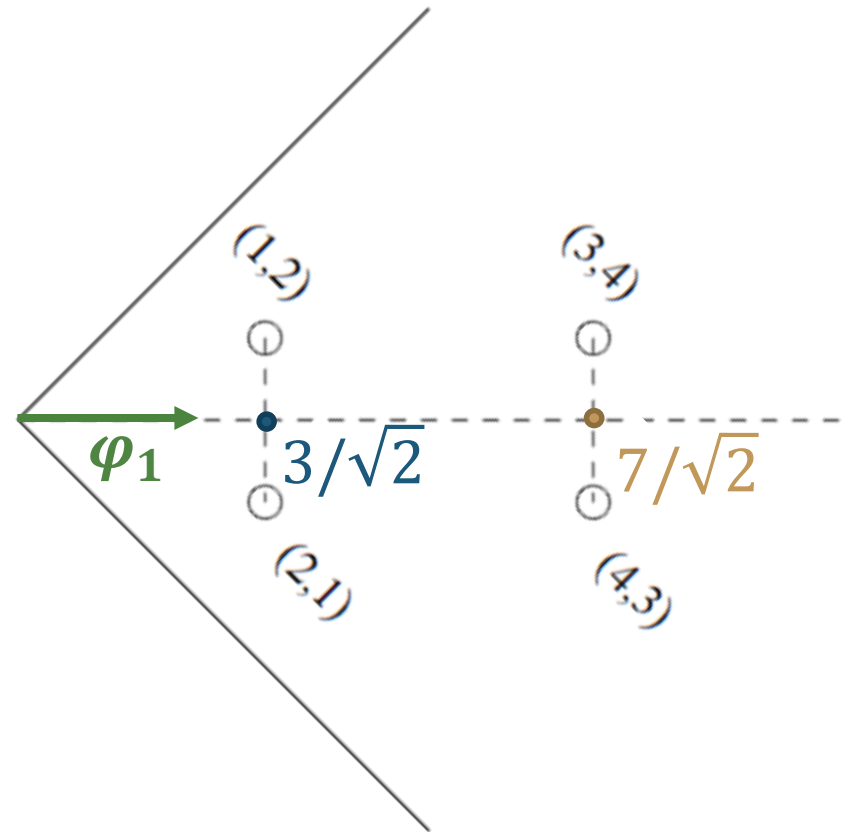
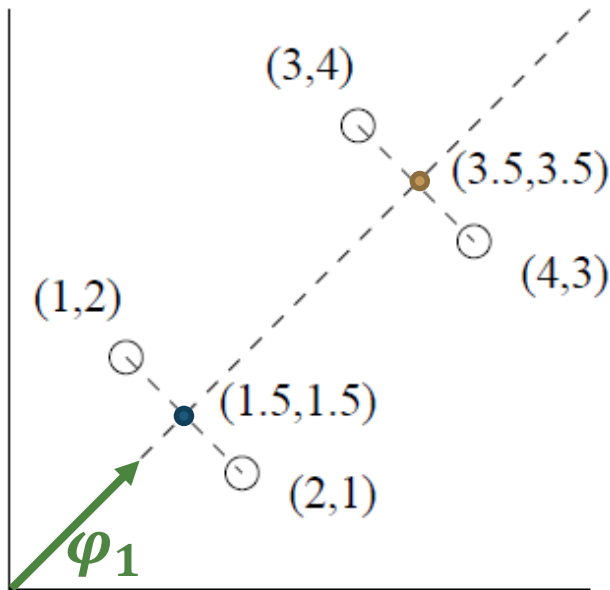
- Die empirische Varianz von $Z_{.1} = \varphi_{11} A_{.1} + \varphi_{21} A_{.2}$ ist maximal für $\varphi_{11} = \frac{1}{\sqrt{2}}$ und $\varphi_{21} = \frac{1}{\sqrt{2}}$

- Somit $Z_1 = \begin{pmatrix} 3/\sqrt{2} \\ 3/\sqrt{2} \\ 7/\sqrt{2} \\ 7/\sqrt{2} \end{pmatrix}$ und $\boldsymbol{\varphi}_1 = \begin{pmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix}$ gibt die neue x-Achse



PCA: Beispiel

- Somit $Z_1 = \begin{pmatrix} 3/\sqrt{2} \\ 3/\sqrt{2} \\ 7/\sqrt{2} \\ 7/\sqrt{2} \end{pmatrix}$ und $\varphi_1 = \begin{pmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix}$ gibt die neue x-Achse



PCA: Berechnung

- Der Ladungsvektor $\boldsymbol{\varphi}_1$ ist der Eigenvektor des größten Eigenwerts der Matrix $A^T A$ (Kovarianzmatrix von A)
- Beispiel: $A^T A = \begin{pmatrix} 30 & 28 \\ 28 & 30 \end{pmatrix}$
- Sei M eine quadratische Matrix. Eine reelle Zahl λ heißt *Eigenwert* von M und ein Vektor $e \neq 0$ der dazugehörige **Eigenvektor**, falls

$$Me = \lambda e.$$

- Beispiel:

$$\begin{bmatrix} 30 & 28 \\ 28 & 30 \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} = 58 \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$$

Eigenwerte und -vektoren

- Berechnung der Eigenvektoren und –werte einer Matrix M z.B. über die **Power-Iteration-Methode**:
 - Zu Beginn: beliebiger Vektor $x_0 \neq 0$
 - Iteration: $x_{k+1} = \frac{Mx_k}{||Mx_k||}$, wobei $|| \dots ||$ die euklidische Norm $||v|| = \sqrt{\sum_i v_i^2}$
 - Stopp, falls Änderungen in x_k vernachlässigbar klein
- Beispiel: $M = \begin{bmatrix} 30 & 28 \\ 28 & 30 \end{bmatrix}$ und $x_0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$
 - $Mx_0 = \begin{bmatrix} 30 & 28 \\ 28 & 30 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 58 \\ 58 \end{bmatrix}$ und $||Mx_0|| = \sqrt{58^2 + 58^2} = 82.02$
 - $x_1 = \begin{bmatrix} 0.707 \\ 0.707 \end{bmatrix}$
 - $Mx_1 = \begin{bmatrix} 30 & 30 \\ 28 & 28 \end{bmatrix} \begin{bmatrix} 0.707 \\ 0.707 \end{bmatrix} = \begin{bmatrix} 41.01 \\ 41.01 \end{bmatrix}$ und $||Mx_1|| = 57.997$
 - $x_2 = \begin{bmatrix} 0.707 \\ 0.707 \end{bmatrix} \approx \begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix}$

Eigenwerte und -vektoren

- Die Power-Iteration-Methode berechnet den ersten Eigenvektor \mathbf{x} (mit dem größten Eigenwert)
- Dazugehörige Eigenwert: $\lambda = \mathbf{x}^T M \mathbf{x}$
- Beispiel:

$$\begin{bmatrix} 0.707 \\ 0.707 \end{bmatrix}^T \begin{bmatrix} 30 & 28 \\ 28 & 30 \end{bmatrix} \begin{bmatrix} 0.707 \\ 0.707 \end{bmatrix} \approx 58$$

- Reduzierung der Matrix M um den Anteil, der durch den ersten Eigenwert und –vektor generiert wird:

$$M^* := M - \lambda \mathbf{x} \mathbf{x}^T$$

- Power-Iteration-Methode auf M^* berechnet den zweiten Eigenvektor von M (mit dem zweitgrößten Eigenwert von M)
- Analoges Vorgehen für die Berechnung weiterer Eigenvektoren/-werte

PCA: zweite Komponente

- Der zweite Eigenvektor von $A^T A$ entspricht dem Ladungsvektor $\boldsymbol{\varphi}_2$ der zweiten Hauptkomponente von A

- Die *zweite Hauptkomponente* von A ist gegeben durch den Vektor

$$Z_{.2} = \varphi_{12} A_{.1} + \varphi_{22} A_{.2} + \cdots + \varphi_{n2} A_{.n}$$

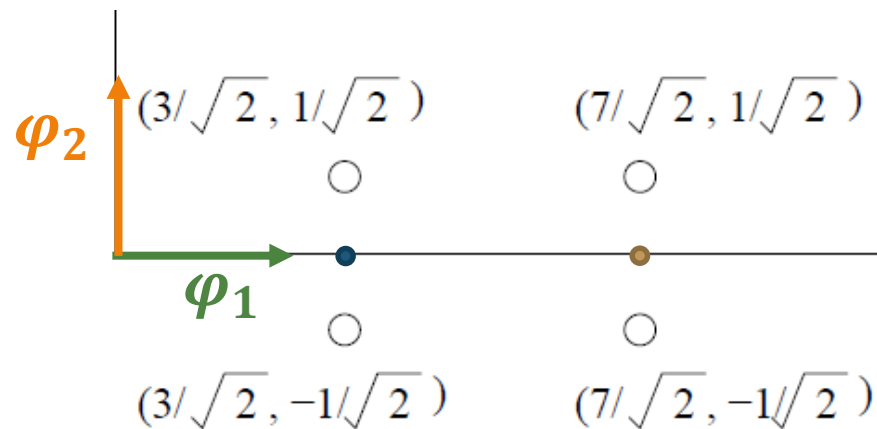
mit $\sum_{j=1}^n \varphi_{j2}^2 = 1$ und maximaler empirischer Varianz $\frac{1}{m} \sum_{i=1}^m Z_{i2}^2$

- Außerdem muss gelten, dass $Z_{.1}$ und $Z_{.2}$ *unkorreliert* sind, d.h.

$$\sum_{i=1}^m Z_{i1} \cdot Z_{i2} = 0$$

- Beispiel:

$$Z_{.2} = \begin{pmatrix} 1/\sqrt{2} \\ -1/\sqrt{2} \\ 1/\sqrt{2} \\ -1/\sqrt{2} \end{pmatrix} \text{ und } \boldsymbol{\varphi}_2 = \begin{pmatrix} -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}$$



PCA: Hinweise

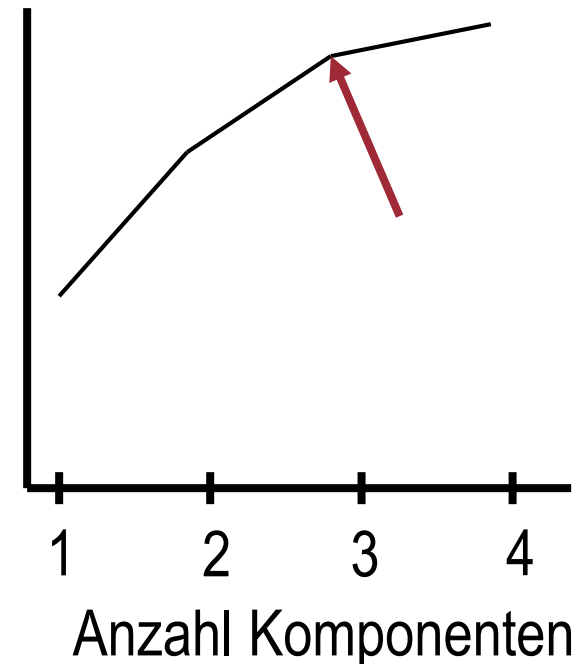
- Spalten sollten gleiche Skalierung haben: neben Zentrierung des Mittelwertes auch gleiche Standardabweichung
- Anteil der Varianz, welcher durch die k-te Hauptkomponenten erklärt wird:

$$PVE_k = \frac{\sum_{i=1}^m Z_{ik}^2}{\sum_{j=1}^n \sum_{i=1}^m A_{ij}^2}$$

- Kumulierter Anteil der erklärten Varianz:

$$\sum_k PVE_k$$

- Wähle die Anzahl der Hauptkomponenten so, dass der kumulierte Anteil der erklärten Varianz durch eine weitere Komponente *nicht stark* ansteigt



PCA: Beispiel

Rekonstruktion eines Bildes über die wichtigsten Hauptkomponenten



800 x 644
Pixel

0.914	0.914	0.914	0.910	...
0.929	0.929	0.925	0.918	...
0.910	0.910	0.902	0.894	...
0.906	0.902	0.898	0.894	...
0.898	0.894	0.890	0.866	...
...

Quelle: <https://kieranhealy.org/blog/archives/2019/10/27/reconstructing-images-using-pca/>

PCA: Beispiel

Hauptkomponententanalyse auf Spalten der Pixel-Matrix:

k	1	2	3	4	5	6	7	8	9
$\sum_k PVE_k$	0.37	0.56	0.65	0.70	0.73	0.76	0.79	0.81	0.83

Recovering the content of an 800x600 pixel image
from a Principal Components Analysis of its pixels

First 2 Components



First 3 Components



y

First 10 Components



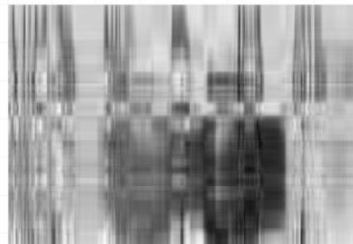
First 20 Components



First 4 Components



First 5 Components



First 50 Components



First 100 Components



Quelle: <https://kieranhealy.org/blog/archives/2019/10/27/reconstructing-images-using-pca/>

Inhaltsverzeichnis

- Einführung
- Hauptkomponentenanalyse
- **Singulärwertzerlegung**
- CUR-Zerlegung

Singulärwertzerlegung

- PCA mit k Hauptkomponenten:
 - Mit $\mathbf{Z} := (Z_{.1} \quad Z_{.2} \quad \dots \quad Z_{.k})$, und
 - $\boldsymbol{\varphi} := (\boldsymbol{\varphi}_1 \quad \boldsymbol{\varphi}_2 \quad \dots \quad \boldsymbol{\varphi}_k)$
 - gilt:

$$\mathbf{Z}_{[m \times k]} = \mathbf{A}_{[m \times n]} \cdot \boldsymbol{\varphi}_{[n \times k]}$$

- Angenommen, wir finden eine Zerlegung von \mathbf{A} der Form

$$\mathbf{A}_{[m \times n]} = \mathbf{B}_{[m \times k]} \cdot (\boldsymbol{\varphi}^T)_{[k \times n]}$$

- Dann gilt

$$\mathbf{Z}_{[m \times k]} = \mathbf{B}_{[m \times k]} \cdot \boldsymbol{\varphi}_{[k \times n]}^T \cdot \boldsymbol{\varphi}_{[n \times k]} = \mathbf{B}_{[m \times k]}$$

Singulärwertzerlegung

- Singular Value Decomposition (**SVD**)
- Es existiert eine Zerlegung einer Matrix A in das Produkt dreier Matrizen:

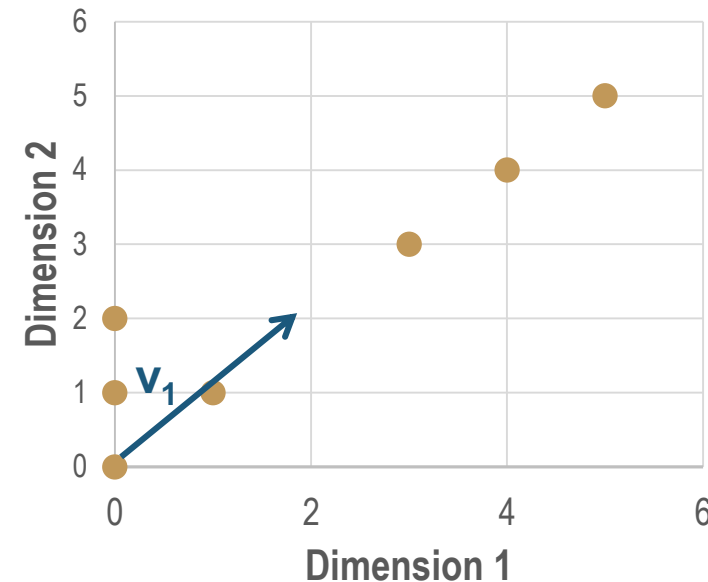
$$\begin{matrix} & n \\ & \boxed{A} \\ m & \end{matrix} = \begin{matrix} & r \\ & \boxed{U} \\ m & \end{matrix} \times \begin{matrix} & r \\ & \boxed{\Sigma} \\ & \end{matrix} \times \begin{matrix} & n \\ & \boxed{V^T} \\ & r \end{matrix}$$

$$A_{[m \times n]} = U_{[m \times r]} \cdot \Sigma_{[r \times r]} \cdot (V_{[n \times r]})^T$$

- wobei Σ eine Diagonalmatrix mit nicht-negativen Einträgen ist (als **Singulärwerte** bezeichnet),
- die Spalten von U und V orthonormal sind (d.h. $U^T \cdot U = V^T \cdot V = I$)
- und $r = \text{Rang von } A$ (Anzahl der Faktoren)

Beispiel

- Matrix V gibt die Faktoren
- Abbildung zeigt 2-dimensionale Projektion der Datenpunkte



$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 3 & 3 & 3 & 0 & 0 \\ 4 & 4 & 4 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 2 & 0 & 4 & 4 \\ 0 & 0 & 0 & 5 & 5 \\ 0 & 1 & 0 & 2 & 2 \end{bmatrix} = \begin{bmatrix} 0.13 & 0.02 & -0.01 \\ 0.41 & 0.07 & -0.03 \\ 0.55 & 0.09 & -0.04 \\ 0.68 & 0.11 & -0.05 \\ 0.15 & -0.59 & 0.65 \\ 0.07 & -0.73 & -0.67 \\ 0.08 & -0.29 & 0.32 \end{bmatrix}$$

$$\mathbf{X} \begin{bmatrix} 12.4 & 0 & 0 \\ 0 & 9.5 & 0 \\ 0 & 0 & 1.3 \end{bmatrix} \mathbf{X}$$

$$\mathbf{v}_1 \begin{bmatrix} 0.56 & 0.59 & 0.56 & 0.09 & 0.09 \\ 0.12 & -0.02 & 0.12 & -0.69 & -0.69 \\ 0.40 & -0.80 & 0.40 & 0.09 & 0.09 \end{bmatrix}$$

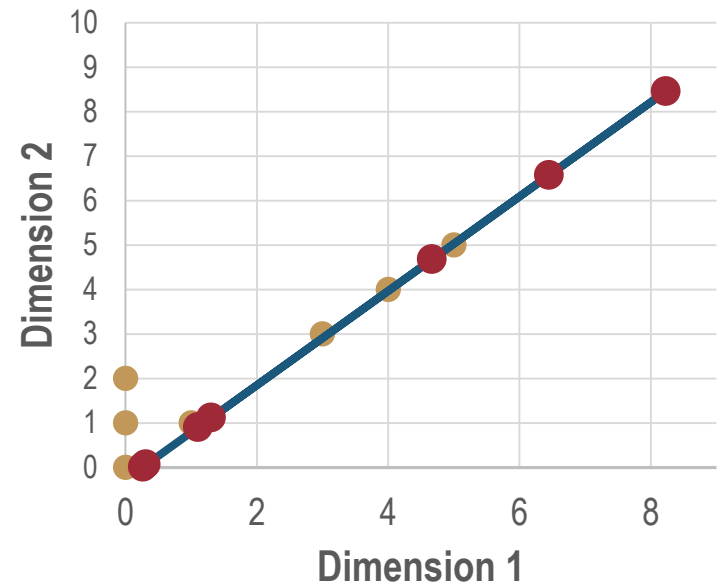
Beispiel

$$\begin{bmatrix} 0.13 & 0.02 & -0.01 \\ 0.41 & 0.07 & -0.03 \\ 0.55 & 0.09 & -0.04 \\ 0.68 & 0.11 & -0.05 \\ 0.15 & -0.59 & 0.65 \\ 0.07 & -0.73 & -0.67 \\ 0.07 & -0.29 & 0.32 \end{bmatrix} \mathbf{x}$$

$$\begin{bmatrix} 12.4 & 0 & 0 \\ 0 & 9.5 & 0 \\ 0 & 0 & 1.3 \end{bmatrix}$$

$$= \begin{bmatrix} 1.71 & 0.22 & -0.01 \\ 5.15 & 0.67 & -0.04 \\ 6.87 & 0.90 & -0.06 \\ 8.58 & 1.12 & -0.07 \\ 1.90 & -5.62 & 0.88 \\ 0.90 & -6.95 & -0.91 \\ 0.95 & -2.81 & 0.44 \end{bmatrix}$$

Projektionen auf die Achse
des ersten Faktors



Beispiel: Interpretation

U ist Nutzer-Faktor-Matrix

Romantik \longrightarrow SciFi \longleftarrow

Nutzer

Matrix	Alien	Serenity	Casablanca	Amelie
1	1	1	0	0
3	3	3	0	0
4	4	4	0	0
5	5	5	0	0
0	2	0	4	4
0	0	0	5	5
0	1	0	2	2

SciFi-Faktor

Romantik-Faktor

Stärke des SciFi-Faktors

$$\begin{bmatrix} 0.13 & 0.02 & -0.01 \\ 0.41 & 0.07 & -0.03 \\ 0.55 & 0.09 & -0.04 \\ 0.68 & 0.11 & -0.05 \\ 0.15 & -0.59 & 0.65 \\ 0.07 & -0.73 & -0.67 \\ 0.07 & -0.29 & 0.32 \end{bmatrix} \times \begin{bmatrix} 12.4 & 0 & 0 \\ 0 & 9.5 & 0 \\ 0 & 0 & 1.3 \end{bmatrix} \times$$

V ist Film-Faktor-Matrix

0.56	0.59	0.56	0.09	0.09
0.12	-0.02	0.12	-0.69	-0.69
0.40	-0.80	0.40	0.09	0.09

SVD: Dimensionsreduktion

Reduktion der Dimensionen, falls Spaltenrang kleiner als n

Zusätzliche Reduktion: Setze kleinsten Singulärwerte auf Null

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 3 & 3 & 3 & 0 & 0 \\ 4 & 4 & 4 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 2 & 0 & 4 & 4 \\ 0 & 0 & 0 & 5 & 5 \\ 0 & 1 & 0 & 2 & 2 \end{bmatrix} \approx \begin{bmatrix} 0.13 & 0.02 & -0.01 \\ 0.41 & 0.07 & -0.03 \\ 0.55 & 0.09 & -0.04 \\ 0.68 & 0.11 & -0.05 \\ 0.15 & -0.59 & 0.65 \\ 0.07 & -0.73 & -0.67 \\ 0.07 & -0.29 & 0.32 \end{bmatrix} \mathbf{X} \begin{bmatrix} 12.4 & 0 & 0 \\ 0 & 9.5 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{X} \begin{bmatrix} 0.56 & 0.59 & 0.56 & 0.09 & 0.09 \\ 0.12 & -0.02 & 0.12 & -0.69 & -0.69 \\ 0.40 & -0.80 & 0.40 & 0.09 & 0.09 \end{bmatrix}$$

Rang-2-Approximation von A (je größer der Rang desto genauer die Approximation)

SVD: Dimensionsreduktion

$$\underbrace{\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 3 & 3 & 3 & 0 & 0 \\ 4 & 4 & 4 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 2 & 0 & 4 & 4 \\ 0 & 0 & 0 & 5 & 5 \\ 0 & 1 & 0 & 2 & 2 \end{bmatrix}}_A \approx \begin{bmatrix} 0.13 & 0.02 & -0.01 \\ 0.41 & 0.07 & -0.03 \\ 0.55 & 0.09 & -0.04 \\ 0.68 & 0.11 & -0.05 \\ 0.15 & -0.59 & 0.65 \\ 0.07 & -0.73 & -0.67 \\ 0.07 & -0.29 & 0.32 \end{bmatrix} \mathbf{X} \begin{bmatrix} 12.4 & 0 & 0 \\ 0 & 9.5 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{X} \begin{bmatrix} 0.56 & 0.59 & 0.56 & 0.09 & 0.09 \\ 0.12 & -0.02 & 0.12 & -0.69 & -0.69 \\ 0.40 & -0.80 & 0.40 & 0.09 & 0.09 \end{bmatrix}$$

$$\underbrace{\begin{bmatrix} 0.92 & 0.95 & 0.92 & 0.01 & 0.01 \\ 2.91 & 3.01 & 2.91 & -0.01 & -0.01 \\ 3.90 & 4.04 & 3.90 & 0.01 & 0.01 \\ 4.82 & 5.00 & 4.82 & 0.03 & 0.03 \\ 0.70 & 0.53 & 0.70 & 4.11 & 4.11 \\ -0.69 & 1.34 & -0.69 & 4.78 & 4.78 \\ 0.32 & 0.23 & 0.32 & 2.01 & 2.01 \end{bmatrix}}_B =$$

Genauigkeit über
Frobeniusnorm:

$$\|A - B\|_F$$

$$= \sqrt{\sum_{ij} (A_{ij} - B_{ij})^2}$$

SVD

- **Satz:** Sei k mit $0 \leq k \leq r$ die Anzahl der gewünschten Faktoren, $\mathbf{A} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T$ und $\mathbf{B} = \mathbf{U} \mathbf{S} \mathbf{V}^T$ wobei \mathbf{S} aus $\mathbf{\Sigma}$ konstruiert wurde, indem die letzten $r - k$ Diagonalelemente auf Null gesetzt wurden. Dann gilt:

$$\mathbf{B} = \underset{\mathbf{B}}{\operatorname{argmin}} \|\mathbf{A} - \mathbf{B}\|_F$$

- Für eine gegebene Anzahl an Faktoren k minimiert SVD den Fehler $\|\mathbf{A} - \mathbf{B}\|_F$, so dass \mathbf{B} die *beste* Rang- k -Approximation für \mathbf{A} darstellt
- Wie klein sollte man k wählen?
- Behalte 80-90% der „Energie“ $\sum_i \sigma_i^2$ (Summe über die quadrierten Diagonalelemente von $\mathbf{\Sigma}$)
- Beispiel: Singulärwerte 12.4, 9.5, und 1.3 \rightarrow Energie: 245.7
 - Entfernen des letzten Singulärwertes setzt Energie auf 244 (99%)
 - Mit nur dem größten Singulärwert wäre die Energie auf 63% reduziert

SVD: Berechnung

- SVD für eine Matrix $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$
- Es gilt: $\mathbf{A}^T = (\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T)^T = (\mathbf{V}^T)^T \mathbf{\Sigma}^T \mathbf{U}^T = \mathbf{V}\mathbf{\Sigma}\mathbf{U}^T$
 - Regel für die Transponierte eines Produkts von Matrizen
 - Zweifache Transposition löst sich auf
 - Transposition einer Diagonalmatrix ergibt die selbe Diagonalmatrix
- Somit gilt: $\mathbf{A}^T \mathbf{A} = \mathbf{V}\mathbf{\Sigma}\mathbf{U}^T \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \mathbf{V}\mathbf{\Sigma}^2 \mathbf{V}^T$
 - Da Spalten von \mathbf{U} orthonormal: $\mathbf{U}^T \mathbf{U} = \mathbf{I}$ (Identitätsmatrix)
 - $\mathbf{\Sigma}^2$ ist eine Diagonalmatrix deren i-tes Diagonalelement das Quadrat des i-ten Diagonalelements von $\mathbf{\Sigma}$ ist
- Da auch die Spalten von \mathbf{V} orthonormal: $\mathbf{A}^T \mathbf{A} \mathbf{V} = \mathbf{V}\mathbf{\Sigma}^2 \mathbf{V}^T \mathbf{V} = \mathbf{V}\mathbf{\Sigma}^2$

D.h. \mathbf{V} ist die Matrix aus Eigenvektoren von $\mathbf{A}^T \mathbf{A}$ und die Diagonalelemente von $\mathbf{\Sigma}^2$ sind die dazugehörigen Eigenwerte.

- Analog: $\mathbf{A} \mathbf{A}^T \mathbf{U} = \mathbf{U}\mathbf{\Sigma}^2$

Inhaltsverzeichnis

- Einführung
- Hauptkomponentenanalyse
- Singulärwertzerlegung
- **CUR-Zerlegung**

CUR-Zerlegung

- Die zu zerlegende Matrix A ist oft spärlich besetzt
- Aber die Matrizen der Zerlegung U und V sind nicht spärlich besetzt, d.h. die Berechnung dieser Matrizen ist sehr aufwendig
- Die CUR-Zerlegung löst dieses Problem
 - Die Matrizen der Zerlegung C und R sind spärlich besetzt, falls A spärlich besetzt
 - Matrix C besteht aus zufällig ausgewählten Spalten von A
 - Matrix R besteht aus zufällig ausgewählten Zeilen von A
 - Wahl von U , so dass $\|A - C \cdot U \cdot R\|_F$ klein

$$\begin{pmatrix} A \end{pmatrix} \approx \begin{pmatrix} C \end{pmatrix} \cdot \begin{pmatrix} U \end{pmatrix} \cdot \begin{pmatrix} R \end{pmatrix}$$

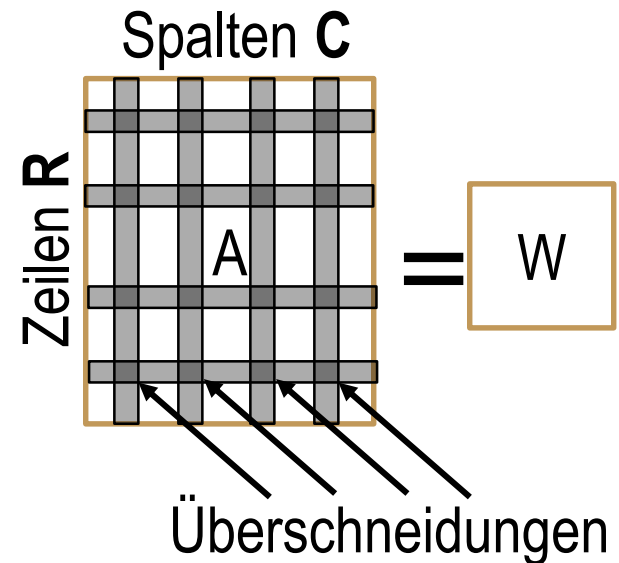
Berechnung der Matrix U

- W sei die Matrix aller Überschneidungen der Spalten C und der Reihen R
- Berechnung der SVD von W : $W = X Z Y^T$
- Dann ist $U := Y (\Sigma^+)^2 X^T$

- Σ^+ : Diagonalmatrix mit Diagonalelementen:

$$\Sigma_{ii}^+ = \frac{1}{\Sigma_{ii}} \text{ falls } \Sigma_{ii} \neq 0 \text{ und } 0 \text{ sonst}$$

- Σ^+ nennt man auch “Moore-Penrose-Pseudoinverse”: Anstatt $\Sigma \Sigma^{-1} = I$ gilt $\Sigma \Sigma^+ \Sigma = \Sigma$ und $\Sigma^+ \Sigma \Sigma^+ = \Sigma^+$



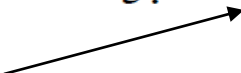
- Beispiel:

$$W = \begin{bmatrix} 0 & 5 \\ 5 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Dann ist:

$$U = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1/5 & 0 \\ 0 & 1/5 \end{bmatrix}^2 \cdot \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1/25 \\ 1/25 & 0 \end{bmatrix}$$

Auswahl der Spalten und Zeilen

- Um den Fehler $\|A - C \cdot U \cdot R\|_F$ deutlich zu verringern sollten die Zeilen und Spalten nach *Wichtigkeit* ausgewählt werden
- Die Wichtigkeit einer Zeile/Spalte: Frobeniusnorm
- Wahrscheinlichkeiten der Auswahl sind proportional zu deren Wichtigkeit
- Beispiel: Spalte [3,4,5] hat Wichtigkeit 50 und die Spalte [3,0,1] hat Wichtigkeit 10 \rightarrow Wahrscheinlichkeit für erste Zeile ist fünfmal so groß wie Wahrscheinlichkeit der zweiten Zeile
- Algorithmus für Spalten: **Input:** matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, sample size c
Output: $\mathbf{C}_d \in \mathbb{R}^{m \times c}$
 1. for $x = 1 : n$ [column distribution]
 2. $P(x) = \sum_i \mathbf{A}(i, x)^2 / \sum_{i,j} \mathbf{A}(i, j)^2$
 3. for $i = 1 : c$ [sample columns]
 4. Pick $j \in 1 : n$ based on distribution $P(x)$
 5. Compute $\mathbf{C}_d(:, i) = \mathbf{A}(:, j) / \sqrt{cP(j)}$
- Eine Spalte kann mehrmals ausgewählt werden
- Skalierung der Spalten, um dies zu korrigieren 

Beispiel

Wkt:

0.012	1	1	1	0	0
0.111	3	3	3	0	0
0.198	4	4	4	0	0
0.309	5	5	5	0	0
0.132	0	0	0	4	4
0.206	0	0	0	5	5
0.033	0	0	0	2	2

Wkt: 0.21 0.185

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 3 & 3 & 3 & 0 & 0 \\ 4 & 4 & 4 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 4 & 4 \\ 0 & 0 & 0 & 5 & 5 \\ 0 & 0 & 0 & 2 & 2 \end{bmatrix} \approx \begin{bmatrix} 1.54 & 0 \\ 4.63 & 0 \\ 6.17 & 0 \\ 7.72 & 0 \\ 0 & 6.58 \\ 0 & 8.22 \\ 0 & 3.29 \end{bmatrix} \times \begin{bmatrix} 0 & 1/25 \\ 1/25 & 0 \end{bmatrix} \times \begin{bmatrix} 0 & 0 & 0 & 7.79 & 7.79 \\ 6.36 & 6.36 & 6.36 & 0 & 0 \end{bmatrix}$$

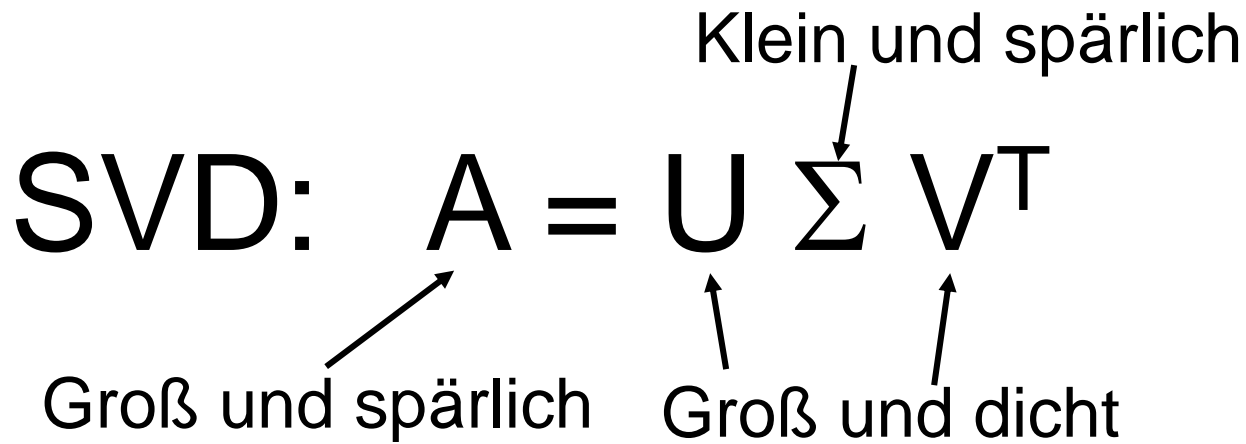
$$= \begin{bmatrix} 0.39 & 0.39 & 0.39 & 0 & 0 \\ 1.18 & 1.18 & 1.18 & 0 & 0 \\ 1.57 & 1.57 & 1.57 & 0 & 0 \\ 1.96 & 1.96 & 1.96 & 0 & 0 \\ 0 & 0 & 0 & 2.05 & 2.05 \\ 0 & 0 & 0 & 2.56 & 2.56 \\ 0 & 0 & 0 & 1.02 & 1.02 \end{bmatrix}$$

SVD vs. CUR

Klein und spärlich

SVD: $A = U \Sigma V^T$

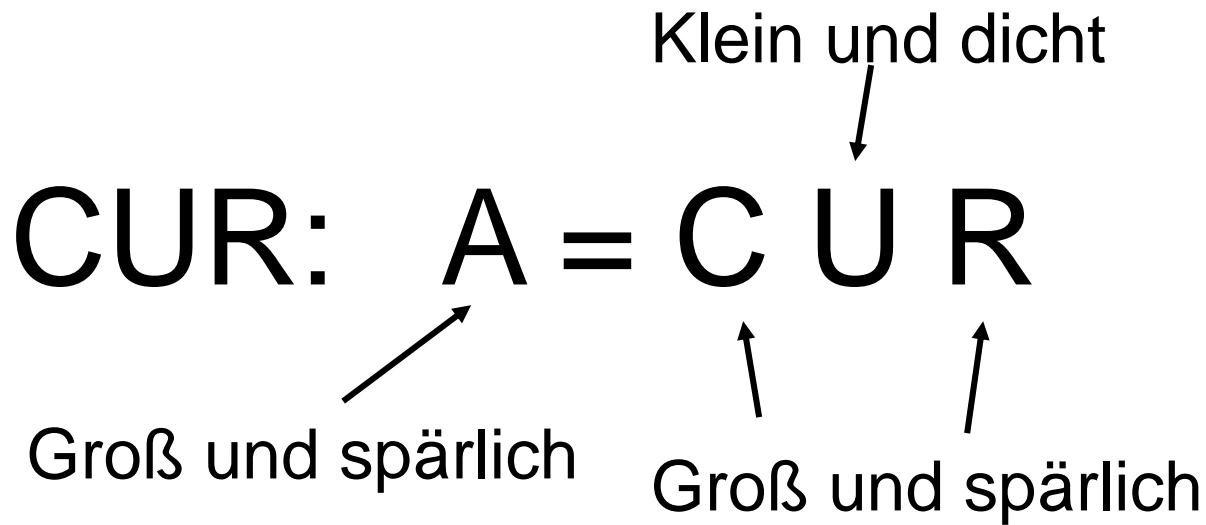
Groß und spärlich Groß und dicht

The diagram shows the SVD decomposition equation A = U Σ V^T. An arrow points from the text 'Klein und spärlich' to the Σ matrix. Two arrows point from the text 'Groß und dicht' to the U and V^T matrices. An arrow points from the text 'Groß und spärlich' to the A matrix.

Klein und dicht

CUR: $A = C U R$

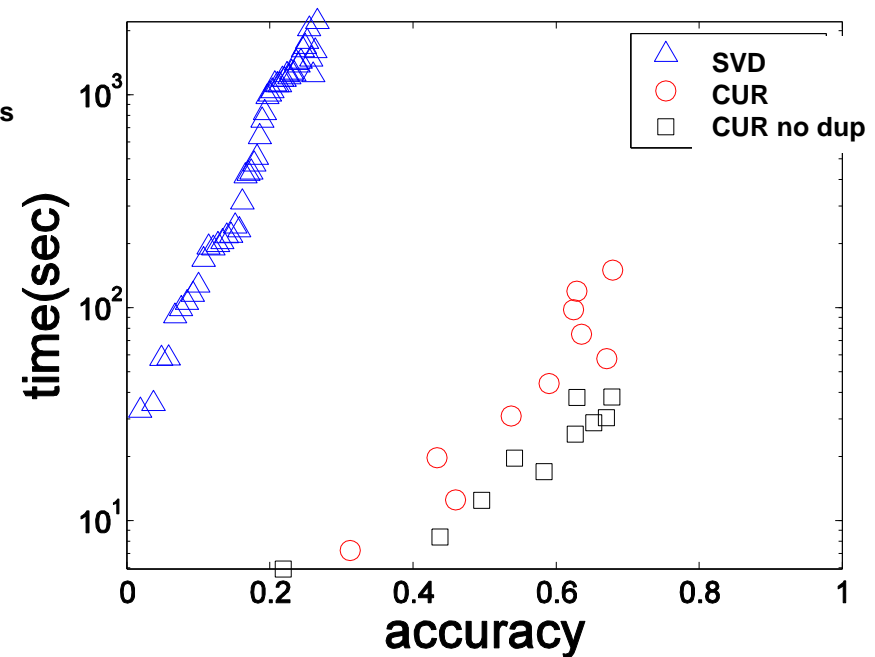
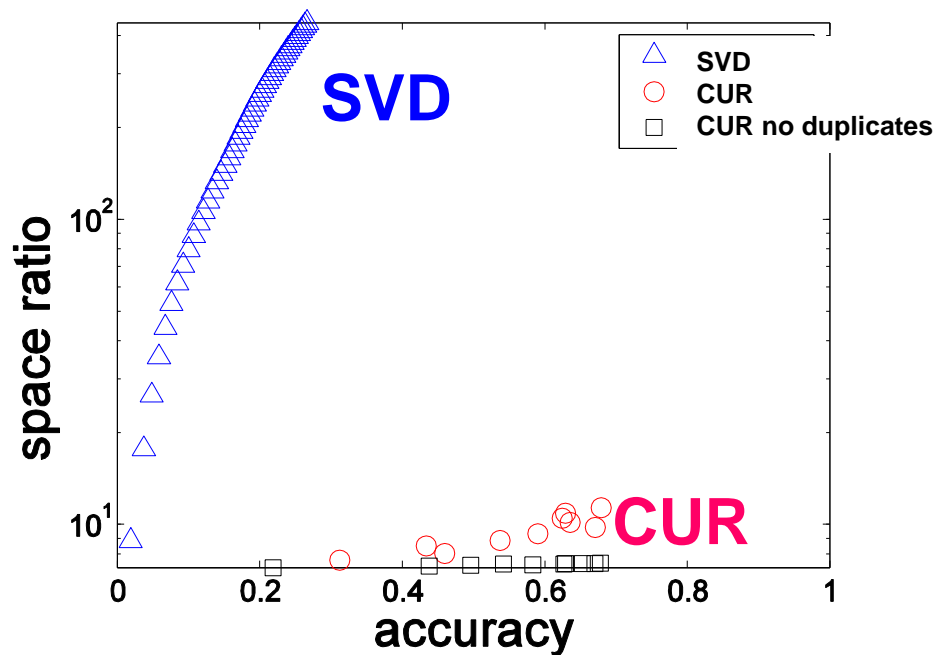
Groß und spärlich Groß und spärlich

The diagram shows the CUR decomposition equation A = C U R. An arrow points from the text 'Klein und dicht' to the U matrix. Two arrows point from the text 'Groß und spärlich' to the C and R matrices. An arrow points from the text 'Groß und spärlich' to the A matrix.

SVD vs. CUR: Experiment

- DBLP Daten
 - Bibliographische Sammlung wissenschaftlicher Publikationen im Bereich Informatik
 - Autor-Konferenz-Matrix A
 - Einträge A_{ij} : Anzahl der Publikationen des Autors i zur Konferenz j
 - 428 000 Autoren (Zeilen), 3 659 Konferenzen (Spalten)
 - Matrix ist sehr groß und spärlich besetzt
- Dimensionsreduktion über SVD und CUR
 - Wie lange laufen die Algorithmen?
 - Wie groß ist der Fehler zwischen approximierter und tatsächlicher Matrix
 - Wie viel Speicherplatz wird benötigt?

SVD vs. CUR: Experiment



- **Accuracy:** $1 - \text{relative Summe der quadrierten Fehler}$
- **Space ratio:** Benötigter Speicherplatz
- **Time:** CPU Zeit

Sun, Faloutsos: *Less is More: Compact Matrix Decomposition for Large Sparse Graphs*, SDM '07.

<http://www.cs.cmu.edu/~jimeng/papers/SunSDM07.pdf>