Singular Value Decomposition
(Singulärwertzerlegung)

Seminar Robust Design

Vitali Müller

Inhalt

- Was ist die SVD?
- Interpretationen
- Anwendungsmöglichkeiten
- Berechnung

Datenmatrix

- Experiment mit n Objekten und m Attributen ergibt n×m-Matrix

- Beispiel:
  - Umfrage, 40 Teilnehmer
  - beurteilen 10 Weine
  - geben jeweils Note von 1 (schlecht) bis 10 (gut)

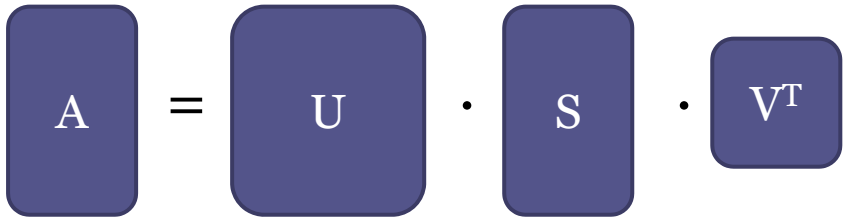
- meist n>>m
- oft n ≈ 1.000.000, m ≈ 100

	W1	W2	…	W10
T1	10	9	…	9
T2	9	8	…	7
T3	6	7	…	6
T4	4	4	…	3
…	…	…	…	…
T39	4	3	…	4
T40	2	3	…	4

Definition

Zerlegung der n×m-Matrix A:

$$A = USV^T$$



mit

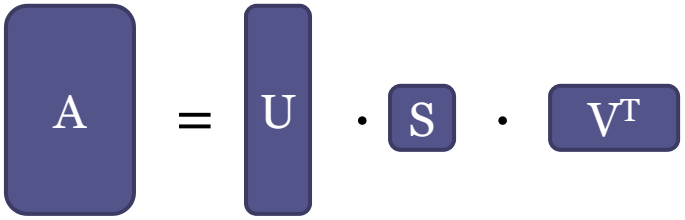
- S n×m-Matrix mit absteigend geordneten Singulärwerten von A auf der Diagonale, sonst 0
- Spalten von U (n×n) und V (m×m) enthalten die Singulärvektoren von A und bilden Orthonormalbasen U und V sind also orthogonal ($UU^T = I$, $VV^T = I$)

Singulärwerte und -vektoren

- Singulärwerte:
  - Wurzeln der Eigenwerte von A^TA
- Singulärvektoren:
  - Eigenvektoren von A^TA (hier Spalten von U)
  - Eigenvektoren von AA^T (hier Spalten von V)

Thin SVD

- m+1-te bis n-te Spalte von U nicht von Bedeutung, da sie mit 0 multipliziert werden (U·S)
- wenn Rang(A)=r, bleiben nur r Spalten von U und V übrig, da es dann nur r Singulärwerte ungleich 0 gibt
- diese verkürzte Schreibweise wird „thin SVD“ genannt



- In der Praxis meistens m=r, da durch Messfehler Attribute fast nie ganz linear abhängig sind

Normalisierung

- Messwerte sollten realistische bzw. vergleichbare Größen aufweisen
- Beispiel:
  - Gewicht und Größe bei Menschen korreliert
  - aber: wenn das Gewicht in Gramm und die Größe in Kilometern angegeben wird, erscheint das Gewicht viel wichtiger ohne Vorkenntnisse

Normalisierung

- meist weiß man aber nicht, was realistisch ist
- Vorgehen deswegen:
  - Annahme: alle Attribute sind gleich wichtig
  - man skaliert alle Einträge (spaltenweise) auf eine ähnliche Größe
  - z.B. indem man jeden Wert einer Spalte durch die Standardabweichung der Spalte dividiert
- außerdem Zentrierung der Werte vorteilhaft:
  - durch Subtraktion des Mittelwerts einer Spalte von allen Einträgen dieser

Interpretationen der SVD

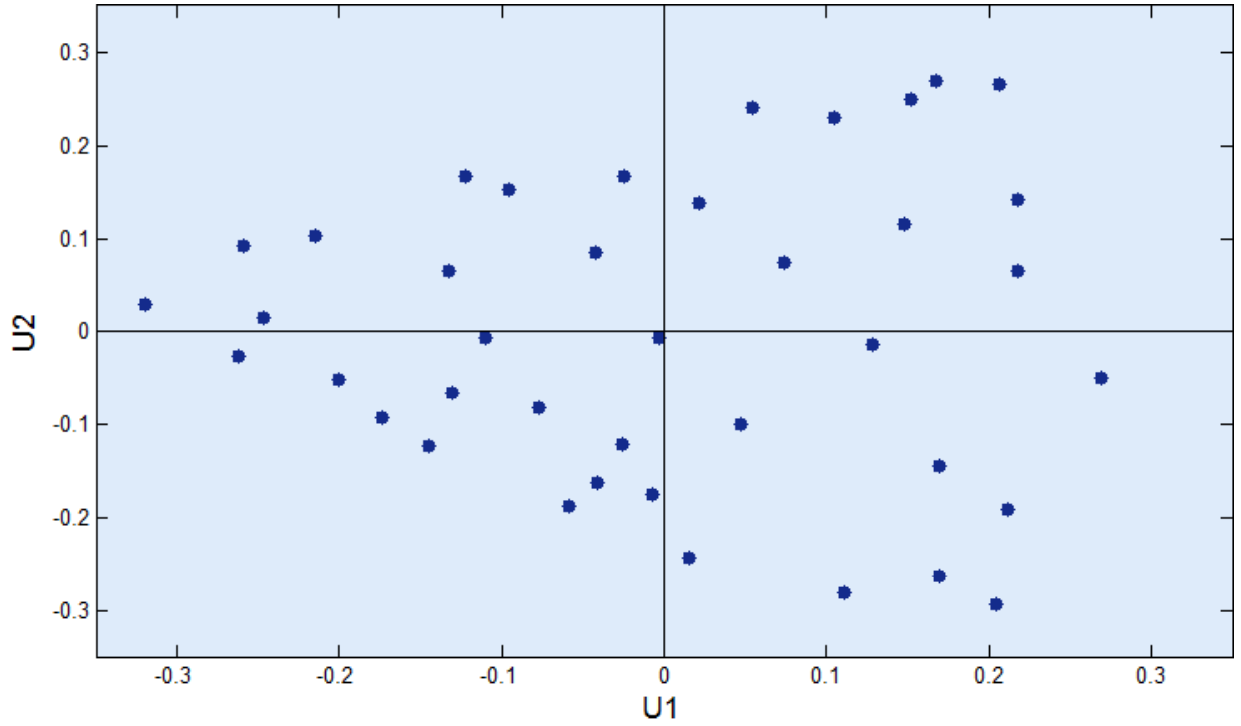
- Aufschlüsselung der Einflussfaktoren oder Prozesse hinter den Daten
- geometrische Interpretation: Objekte als Elemente eines neuen Vektorraums

Einflussfaktoren

- Zeilen von V^T (Spalten von V) können als Einflussfaktoren aufgefasst werden, die den Daten zugrunde liegen
  (sie definieren neue Achsen im $\mathbf{R}^m$)
- Zeilen von U als Koordinaten der Objekte bzgl. dieser Achsen

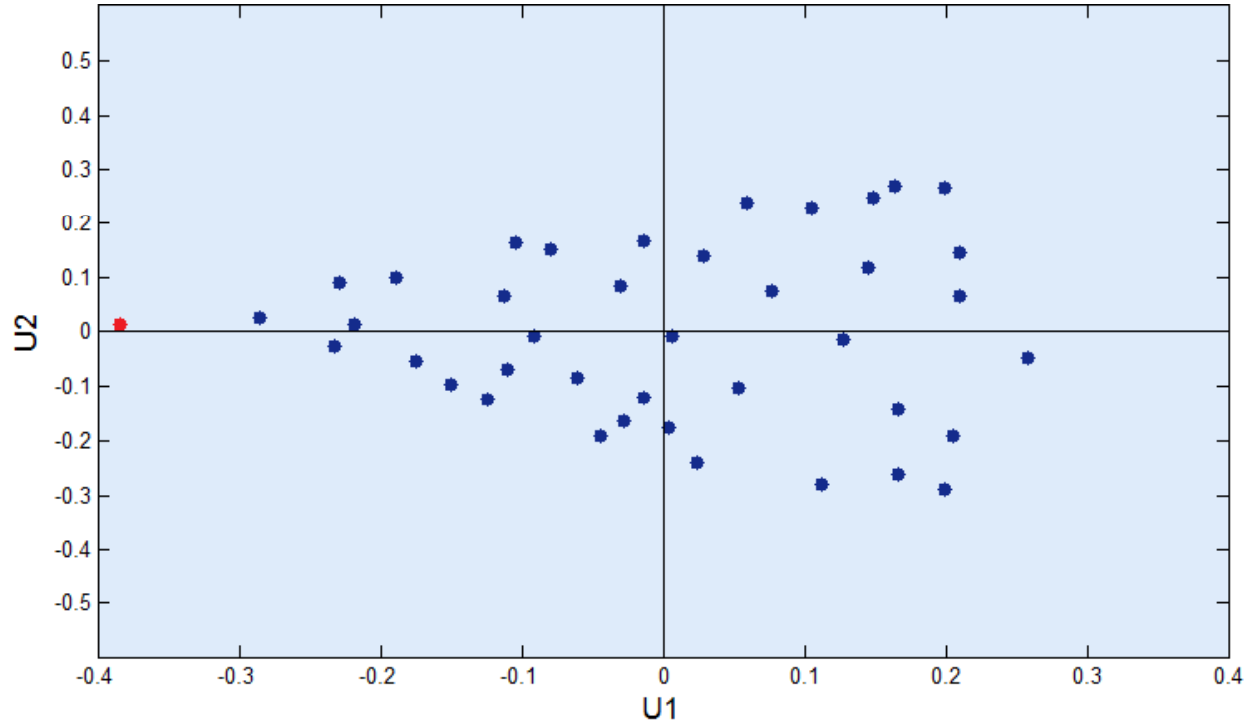
Einflussfaktoren: Beispiel

- Betrachtung der ersten zwei neuen Dimensionen beim Weintest ergibt folgendes Bild
- Punkte entsprechen den Testpersonen (Objekten)



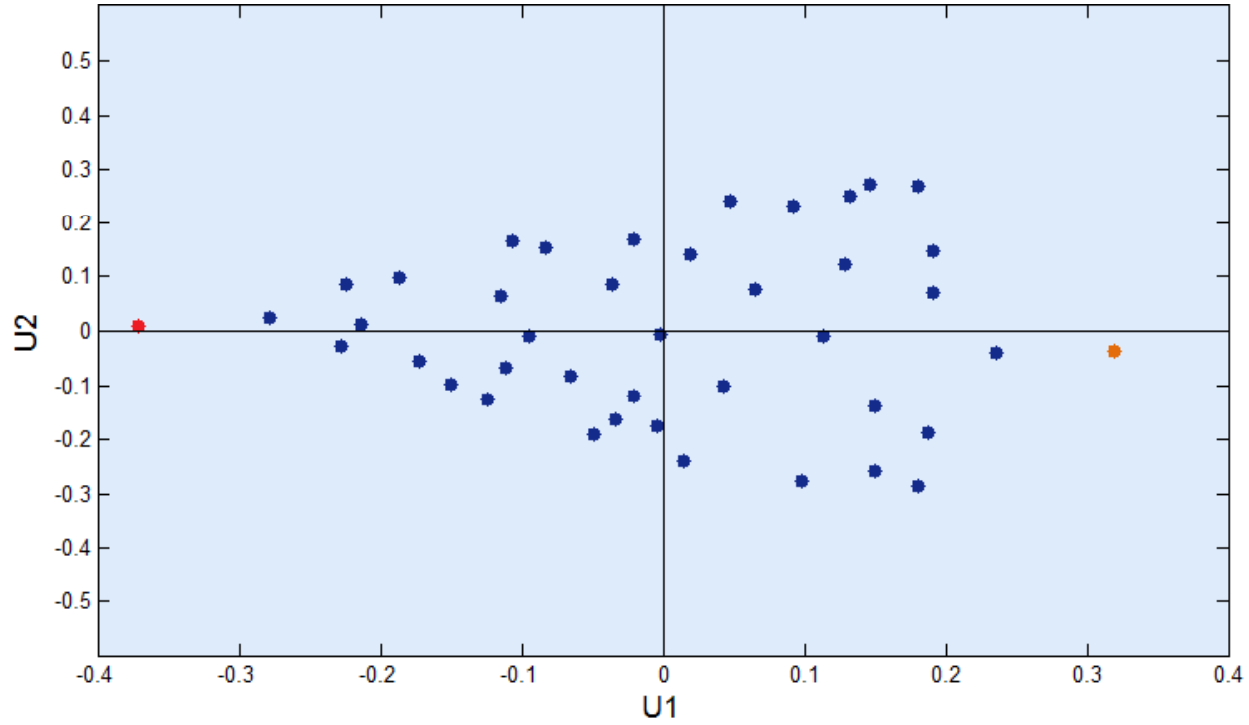
Einflussfaktoren: Beispiel

- Aber was bedeuten diese Dimensionen?
- füge künstliche Objekte mit extremen Werten hinzu:
  (10 10 … 10) (jemand, der jeden Wein sehr gut findet)



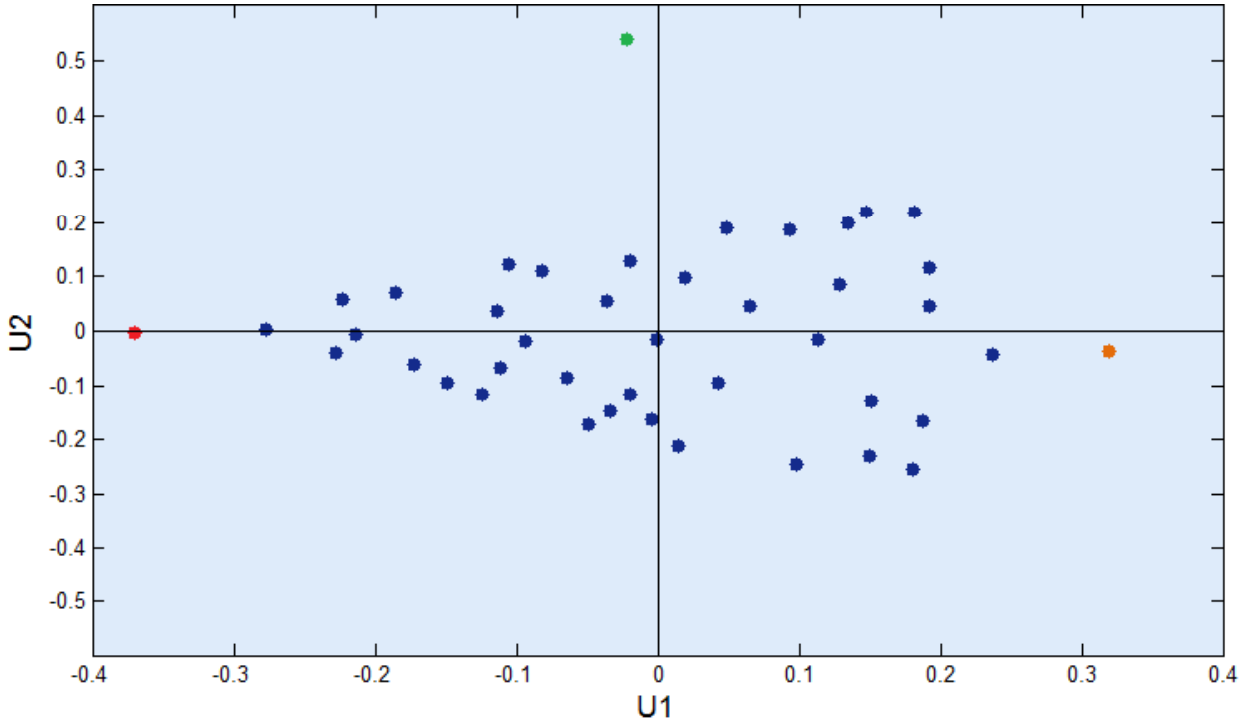
Einflussfaktoren: Beispiel

- Aber was bedeuten diese Dimensionen?
- füge künstliche Objekte mit extremen Werten hinzu:
  (1 1 … 1) (jemand, der jeden Wein sehr schlecht findet)



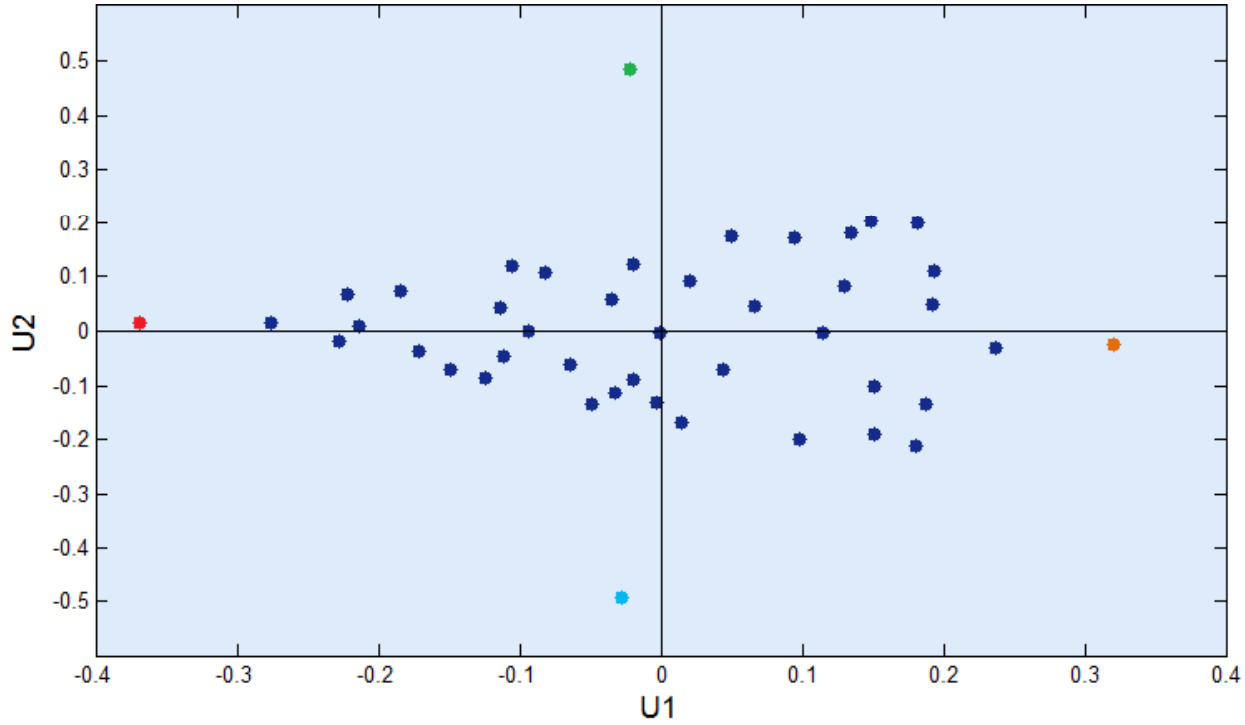
Einflussfaktoren: Beispiel

- Weitere Möglichkeit für künstliche Objekte:
- jemand, der Rotwein sehr mag, aber Weißwein gar nicht: (10 …10 1 … 1)



Einflussfaktoren: Beispiel

- Weitere Möglichkeit für künstliche Objekte:
- jemand, der Rotwein sehr mag, aber Weißwein gar nicht:
  (10 …10 1 … 1) …und umgekehrt: (1 … 1 10 … 10)



Einflussfaktoren: Beispiel

- offensichtliche Ergebnisse:
  - erste neue Dimension U1: Mögen von Wein überhaupt (links: mag Wein, rechts: mag keinen Wein)
  - U2: Präferenz für Rotwein (oben) oder für Weißwein
- Folgerung aus der Dreiecksgestalt: je weniger jemand Wein mag, desto eher hat er eine Vorliebe für Rot- oder Weißwein

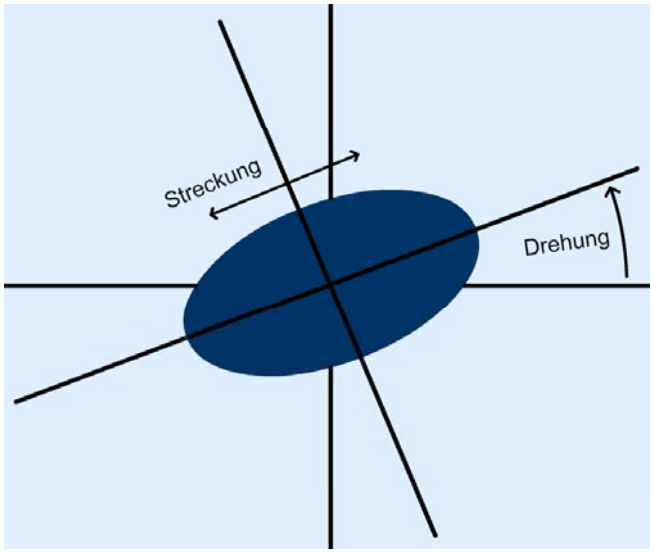
- so kann man bis zu 3 Dimensionen auf einmal graphisch untersuchen
- aber es ist oft nicht so leicht, Faktoren zu deuten

geometrische Interpretation

- Spalten von V^T bilden eine Orthonormalbasis des $\mathbb{R}^m$ (da V orthogonal)
- Zeilen von U als Koordinaten bzgl. dieser Basis
- Einträge von S als Streckungsfaktoren für die einzelnen Achsen
- da die Singulärwerte in S absteigend:
  meiste Variation in der 1. Dimension

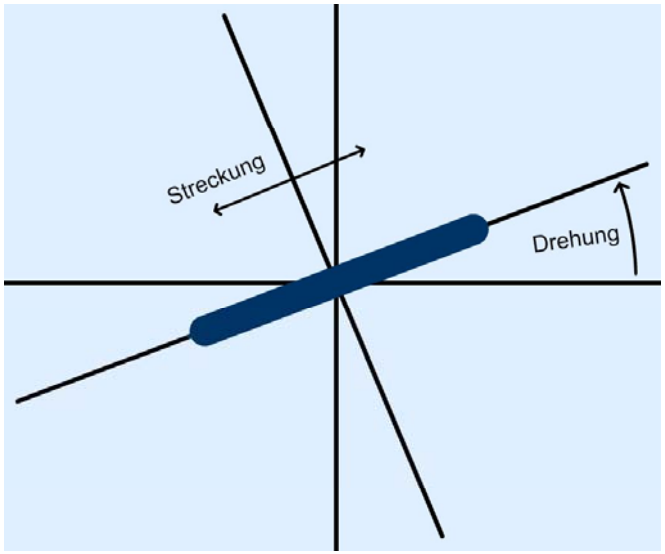
Drehung / Skalierung

- m-dimensionale Einheitskugel
  - wird durch V^T gedreht
  - und durch S skaliert)
  - passt dann „über“ die Daten
- Vorteil der Normalisierung: Daten bilden schon in etwa eine Einheitskugel um 0
  - Drehung/Skalierung konzentriert sich nur auf die Verteilung der Objekte in die jeweiligen Richtungen



Drehung / Skalierung: Spezialfall

- Daten scheinen m-dimensional zu sein, sind aber nur eine r-dimensionale Mannigfaltigkeit
- erkennbar, da weniger neue Koordinaten gebraucht werden, um die Objekte zu beschreiben



- Beispiel: Datenpunkte liegen auf einer Geraden, es werden aber beide Koordinaten genutzt
  - nach SVD: nur noch Ausbreitung entlang der ersten Dimension

Skalarprodukte, Korrelation

- man betrachtet die Datenpunkte als Vektoren
- relative Richtung zweier Vektoren und damit ihr Skalarprodukt geben Auskunft über ihre Korrelation:
  - ungefähr die selbe Richtung bedeutet starke Korrelation und großes positives Skalarprodukt

  - ungefähr entgegengesetzte Richtung: starke negative Korrelation und großes negatives Skalarprodukt

  - Richtungen ungefähr orthogonal: schwache Korrelation und Skalarprodukt nahe bei 0

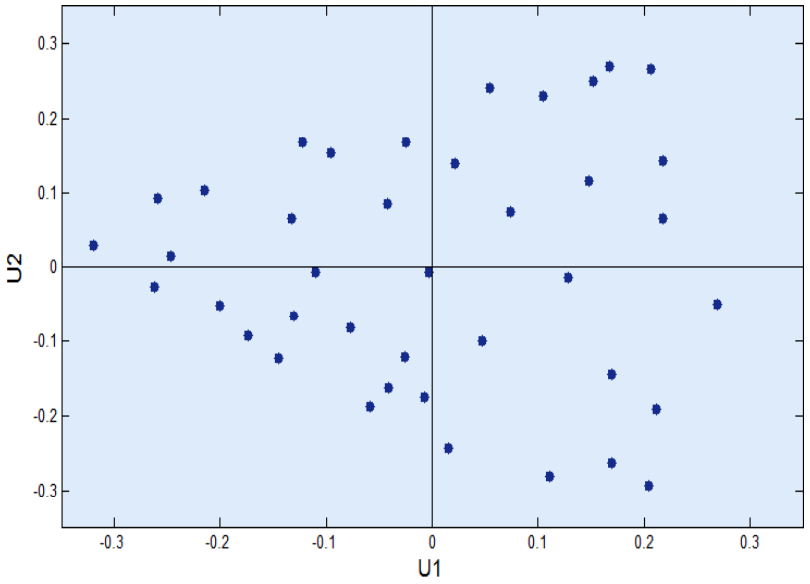
Skalarprodukte, Korrelation

- Objekte, die mit vielen anderen unkorreliert sind, haben kleine Skalarprodukte mit diesen
  - aber: wegen n>>m gibt es nicht genug orthogonale Richtungen
  - einzige Möglichkeit für ein kleines Skalarprodukt: eine Lage nah beim Ursprung,
- Objekte, die mit vielen anderen korreliert sind, „zeigen“ in viele Richtungen, sind also kurz
  - sie liegen also auch nah beim Ursprung
- Objekte, die nur mit einigen anderen korreliert sind:
  - liegen meist weit vom Ursprung entfernt

Skalarprodukte, Korrelation

- Objekte, die mit fast keinen oder mit sehr vielen anderen korreliert sind, sind weniger interessant
- Sortierung nach Abstand vom Ursprung entspricht also der Sortierung nach Interesse für weitere Untersuchungen

- Im Beispiel würde man also zunächst die Objekte in der Nähe der Ecken des Dreiecks untersuchen



A als Summe überlagernder Prozesse

- $A = USV^T$
- $a_{i,j} = u_{i,1}s_{1,1}v_{j,1} + u_{i,2}s_{2,2}v_{j,2} + \ldots + u_{i,m}s_{m,m}v_{j,m}$
- Einträge von A jeweils Summe von m Produkten
- A ergibt sich durch Überlagerung einzelner Prozesse

Anwendungen der SVD

- Auswahl von Daten zur weiteren Analyse
- Datenkompression
- Untersuchung von Korrelationen zwischen Objekten und zwischen Attributen
- Clustering

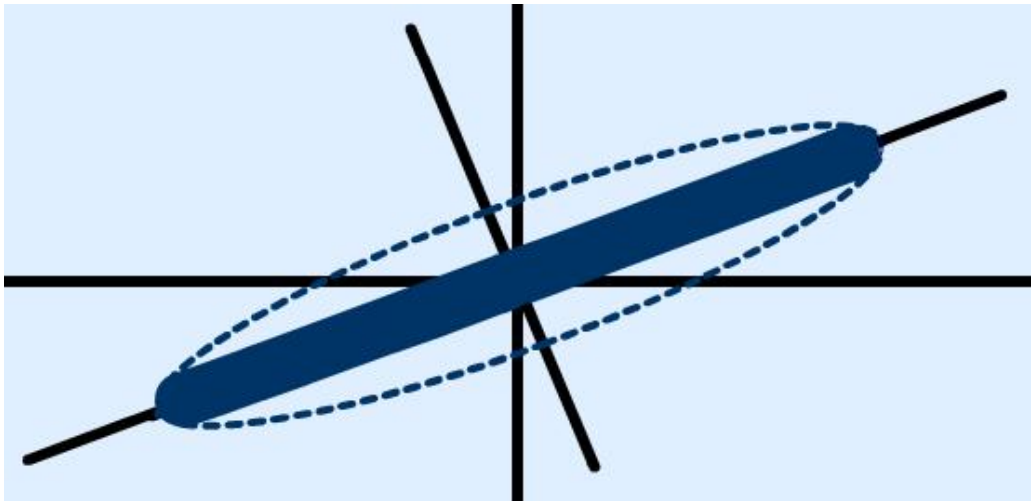
Auswahl von Daten

Im Wesentlichen 2 Gründe, Daten wegzulassen:

- man hält sie für Störungen (noise)
- man will die Prozesse dahinter nicht modellieren (z.B. weil sie vernachlässigbar sind)

Denoising: Beispiel

- 2 perfekt korrelierte Attribute
- SVD würde wirkliche Dimension der Daten „erkennen“
- durch Störungen auch noch nach SVD 2-dimensional
- Streuung in dieser Dimension aber nicht so groß, da keine wirkliche Struktur vorhanden



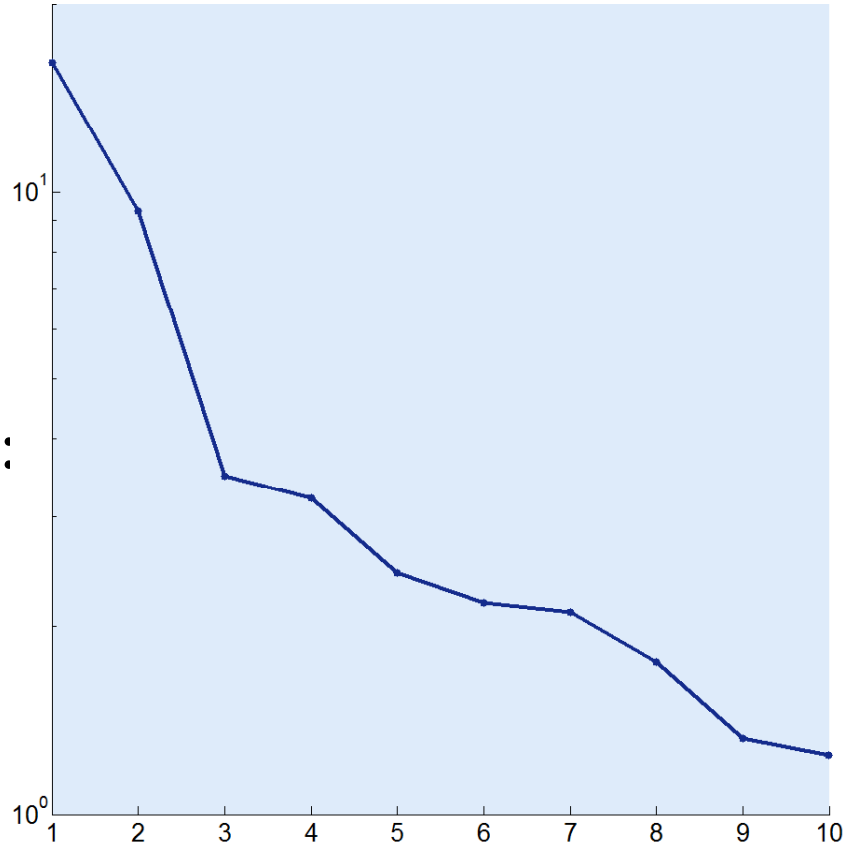
Denoising

- durch Störungen kann die Dimension eines Datensatzes deutlich höher erscheinen als sie in Wirklichkeit ist
- Größe der Singulärwerte gibt Auskunft über Streuung in der jeweiligen Dimension (Streckungsfaktoren)
- Singulärwerte absteigend geordnet, also beinhalten die letzten Dimensionen am ehesten Störungen
- man behält nur die ersten k Dimensionen und lässt den Rest weg

Denoising: Wahl von k

eine der Möglichkeiten:

- Singulärwerte plotten (hier logarithmisch)
- nach einer Abflachung oder einem Knick suchen
- mögliche Entscheidung hier:
  - k=2 (mit Abstand die größten Singulärwerte)
  - k=4 (danach Absenkung)
  - k=7 (so wie bei 4)
- subjektive Entscheidung!



Datenkompression

- man kann aber auch bewusst Daten weglassen, die keine Störungen sind, um Speicherplatz zu sparen
- da die ersten Dimensionen den größten Informationsgehalt haben, behält man diese

Datenkompression

- k Dimensionen behalten bedeutet dabei konkret:
  - (k+1)-te bis m-te Spalte von U gleich 0 setzen → U_k
  - Diagonaleinträge von S ab dem (k+1)-ten 0 setzen → S_k (man speichert nur die k ersten Singulärwerte)
  - (k+1)-te bis m-te Zeile von V^T gleich 0 setzen → V_k^T
  - $A_k = U_k S_k V_k^T$ Approximation für A
  - Rang von A_k ist k
- A_k ist sogar die beste Approximation für A mit Rang k bzgl. der Spektral- und der Frobeniusnorm

Datenkompression

- Speicherung von U_k, V_k und der ersten k Singulärwerte statt A nutzbar zum Sparen von Speicherplatz
- bei n×n-Matrix:
  - U_k und V_k sind n×k-Matrizen
  - 2nk Einträge gegenüber n^2 bei A
  - Ersparnis bei $k<n/2$ um den Faktor $n/2k$
- Anwendung z.B. bei der Speicherung von Bildern:
  - Bild als Matrix
  - Einträge entsprechen Farben der einzelnen Pixel (als Zahlen kodiert)

Datenkompression bei Bildern

Original (400x400)

Approximation durch SVD mit k=200

Datenkompression bei Bildern

Original (400x400)

Approximation durch SVD mit k=150

Datenkompression bei Bildern

Original (400x400)

Approximation durch SVD mit k=100

Datenkompression bei Bildern

Original (400x400)

Approximation durch SVD mit k=50

Datenkompression bei Bildern

Original (400x400)

Approximation durch SVD mit k=5

Datenkompression bei Bildern

- Bilder wurden bearbeitet mit einer Matlab-Routine von utenti.lycos.it/matlab
- Aufruf:

  out=imcompr(‘Input.bmp‘,Rang,‘Output.bmp');

  mit Rang = Anzahl der zu benutzenden Singulärwerte

Korrelationsmatrizen

- Untersuchung von Beziehungen
  - zwischen Objekten: Korrelationsmatrix AA^T
  - zwischen Attributen: Korrelationsmatrix A^TA

Korrelationsmatrizen

- Beispiel:

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 6 \\ -2 & -4 \\ -2 & 1{,}1 \end{bmatrix} \quad AA^T = \begin{bmatrix} 5 & 15 & -10 & 0{,}2 \\ 15 & 45 & -30 & 0{,}6 \\ -10 & -30 & 20 & -0{,}4 \\ 0{,}2 & 0{,}6 & -0{,}4 & 5{,}21 \end{bmatrix}$$

- Objekte 1 und 2 korreliert, 1 und 3 negativ korreliert, 4 ist kaum mit den anderen Objekten korreliert
- Vergleich von $A_k A_k^T$ mit AA^T hilft bei der Unterscheidung zwischen Beziehungen, die durch die ersten k Prozesse bedingt werden und denen, die durch die weggelassenen Prozesse verursacht werden

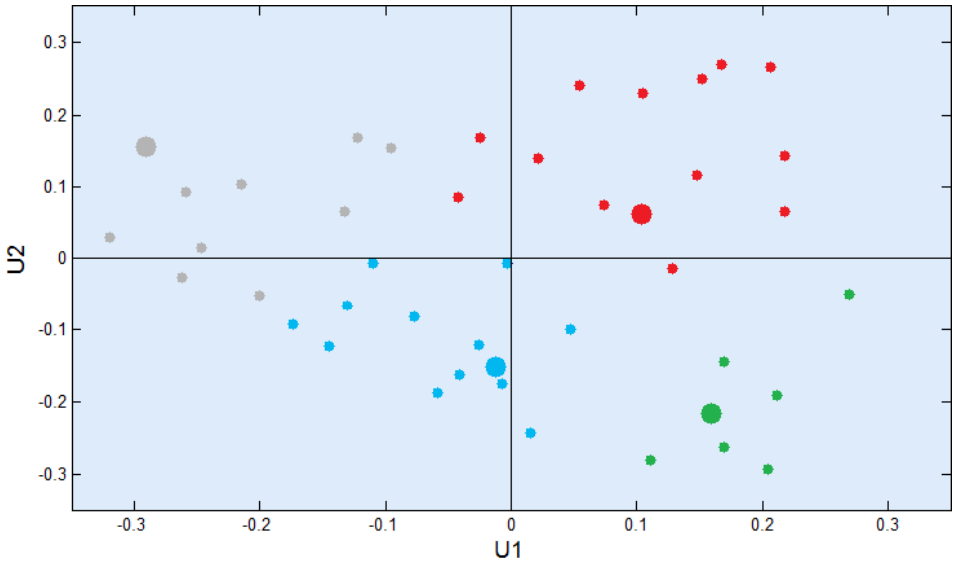
Clustering

- Ziel: Zusammenfassung von ähnlichen Objekten in Gruppen
- Möglichkeiten der Auffassung von Ähnlichkeit:
  - Euklidische Distanz
  - Kosinus-Ähnlichkeit: misst die Korrelation (Winkel)
- Untersuchung der Ähnlichkeit der Objekte nach Anwendung der SVD und Denoising (Zeilen von U) leichter als davor (Zeilen von A), da Dimension deutlich kleiner
- im Folgenden zwei von vielen Clustering-Verfahren

Clustering

k-means-Algorithmus

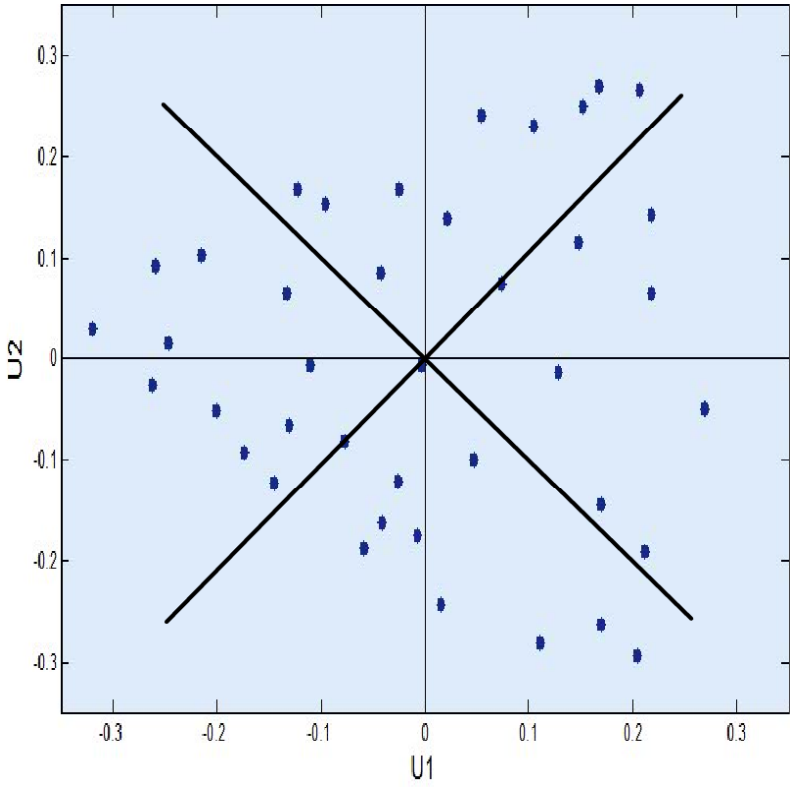
- zufällige Auswahl von k Clusterzentren
- Zuordnung der Objekte zum jeweils nächsten in euklidischer Distanz
- Neuberechnung der Clusterzentren (z.B. Mittelwerte der Koordinaten in jeder Dimension)
- neue Zuordnung der Objekte usw. bis sich nichts mehr ändert



Clustering

Anderer Ansatz :

- 45°-Kegel um Achsen bilden Cluster (1 oder 2 pro Achse)
- jedes Cluster enthält Vektoren, die mit einer bestimmten Achse am stärksten korrelieren
- Kosinus-Ähnlichkeit gegeben



Berechnung der SVD

- direkte Berechnung der Eigenwerte und –vektoren von A^TA numerisch nicht sinnvoll
- viele andere Algorithmen
- Komplexität der SVD im Allgemeinen: n^2m+nm^2
- da m meist viel kleiner als n kann man annehmen: $O(n^2m)$
- in viele Softwarepakete integriert
- in Matlab: [U,S,V]=svd(A)

SVD Aktualisieren

- A bleibt gleich groß, aber einige Werte ändern sich:
  - lässt sich proportional zu n neu berechnen, wenn die Änderungen klein sind
- A bekommt eine zusätzliche Zeile oder Spalte:
  (z.B. künstliches Objekt oder neues Experiment)
  - man benutzt die Gleichung $U=AVS^{-1}$,um die neue Zeile von U zu berechnen
    (Umformung von $A = USV^T$ mit $V^{-1}=V^T$)
  - analog für V
  - schnell zu berechnen, aber Orthogonalität geht verloren

Zusammenfassung

wichtige Anwendungen der SVD:

- Analyse von großen Datenmengen
  - direkt (→ Interpretation als Einflussfaktoren)
  - indirekt durch Verbesserung/Erleichterung anderer Verfahren:
    - Clustering
    - Korrelationsmatrizen
- Filterung von Störungen in den Daten (Denoising)
- Datenkompression (z.B. bei Bildern)