Universität Potsdam
Institut für Informatik
Lehrstuhl Maschinelles Lernen

Sparse PCA

Tobias Scheffer
Michael Großhans
Paul Prasse
Uwe Dick

Vektorraummodell

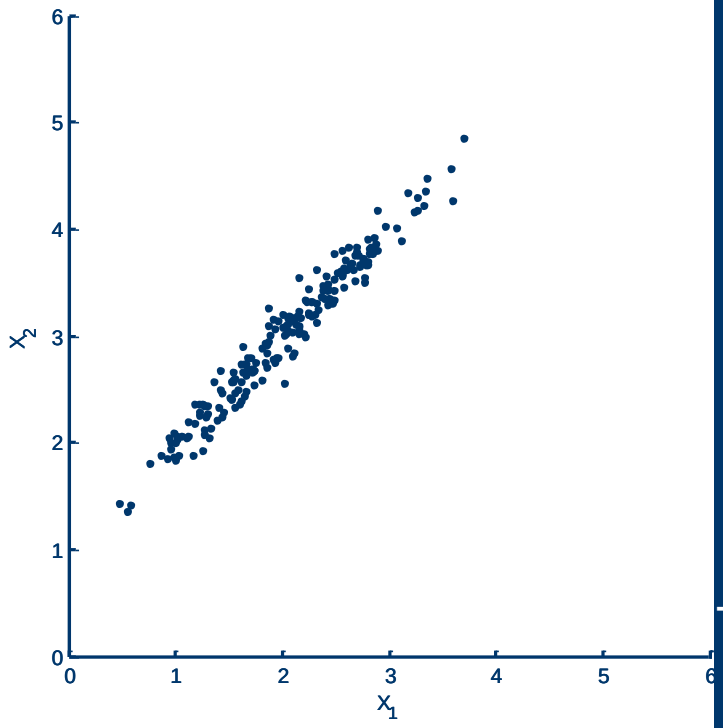
- Jedes Dokument wird als Vektor dargestellt,
  - beispielsweise als binäres Bag-of-Word:
    - An jeder Stelle des Vektors gibt 0/1 an, ob das entsprechende Wort im Text vorhanden ist oder nicht.
  - oder als N-Gram-Modell:
    - 3-Gram-Modell: Jeder Eintrag im Vektor korreliert mit einer Kombination aus 3 Buchstaben (z.b. aaa, aab)
    - An jeder Stelle des Vektors steht die Häufigkeit des Auftretens der entsprechenden Kombination im Text.
- Oft: Hochdimensionale, aber sparse Daten, bspw. Emails:
    - Großes Vokabular (Länge der Vektoren)
    - Wenig Text pro Email (Einträge ungleich 0 pro Vektor)

Vektorraummodell

- Viele Algorithmen können Sparsität der Daten ausnutzen, um effizient Modelle zu bestimmen
  - Laufzeit hängt nicht von der Länge, sondern von der Anzahl an Einträgen ungleich 0 ab.
- Ansonsten führt die hohe Dimensionalität zu hohen Laufzeiten.
  - Ziel: Reduzierung der Anzahl der Dimensionen:
    - Stop-Wörter entfernen, Zahlen entfernen,
    - Seltene Wörter entfernen (z.B. Nutzernamen),
    - Wortstämme nutzen (sein statt bin/ist/sind/usw.),
    - Groß- und Kleinschreibung,
    - uvm.

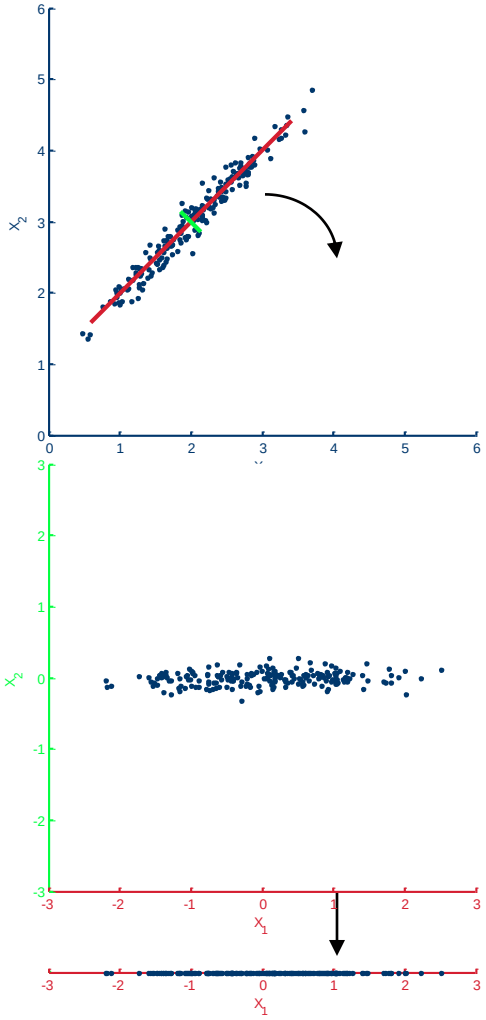
Reduzierung der Dimensionen

- Beispiel:
  - 2-dimensionaler Vektorraum.
  - Rot: Richtung mit Stärkster Streuung der Daten.
  - Grün: Orthogonal zu rot.
  - Vermutung:
    - Rote Komponente reicht aus um Daten zu charakterisieren.
    - Daten sind im Wesentlichen 1-dimensional.



Reduzierung der Dimensionen

- Ansatz (PCA):
  1. Finde Komponenten mit maximaler Streuung iterativ:
     - ★ Jeweils orthogonal zu bisherigen Ausbreitungsrichtungen.
  2. Transformiere Daten in neues Koordinatensystem aufgespannt durch gefundene Komponenten.
  3. Ignoriere Komponenten mit geringer Streuung (hier: grün).
     - ★ Wesentliche Eigenschaften bleiben (hoffentlich!) trotz Reduktion erhalten.



PCA

- Teilprobleme:
  - Bestimmen der Hauptkomponenten.
    - Werden für das Zielkoordinatensystems benötigt.
  - Transformation und Reduktion der Daten.
    - Transformation in das neue Koordinatensystem.
    - Welche Dimensionen können ignoriert werden?
  - Interpretierbarkeit der neuen Daten.
    - Wie kann Interpretierbarkeit der neuen Daten verbessert werden?

Wiederholung: Algebra

- Repräsentationen von Daten
  - Instanz mit *m* Feature: $\mathbf{x} = \left(x_1, \ldots, x_m\right)^{\mathrm{T}}$
  - *n* Instanzen (Datenmatrix): $\mathbf{X} = \left(\mathbf{x}_1, \ldots, \mathbf{x}_n\right)$

- Affin-lineare Transformation von $\mathbb{R}^{m \times n}$ nach $\mathbb{R}^{m' \times n}$
  - Einer Datenmatrix: $A(\mathbf{X}) = \mathbf{A}\left(\mathbf{X} - \mathbf{B}\right)$
  - Reduktion der Feature, wenn $m' < m$

$$\mathbf{A} \in \mathbb{R}^{m' \times m}, \mathbf{B} \in \mathbb{R}^{m \times n}$$

  - Beispiele:
    - ★ Skalierung der Feature durch Diagonalmatrix $\mathbf{A}$
    - ★ Neues Koordinatensystem, wenn Zeilen Orthonormalbasis bilden:
      - für zwei Zeilen $\mathbf{a}_i, \mathbf{a}_j$ gilt $\mathbf{a}_i^{\mathrm{T}} \mathbf{a}_j = \begin{cases} 1 & \text{falls } i = j \\ 0 & \text{falls } i \neq j \end{cases}$

Wiederholung: Algebra
Matrix-Eigenschaften

- Eigenschaften einer Matrix
  - Quadratisch: $n = m$
  - Symmetrisch: $\mathbf{A} = \mathbf{A}^{\mathrm{T}}$
  - Spur (trace): $tr(\mathbf{A}) = \sum_{i=1}^{m} a_{ii}$
  - Rang (rank): $rk(\mathbf{A}) =$ maximale Zahl linear unabhänger Zeilen/Spalten
  - Positiv definit: $\mathbf{A} > 0,$ wenn $\mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{x} > 0 \quad \forall \mathbf{x} \neq \mathbf{0}$ falls $\mathbf{A}$ symmetrisch
  - Positiv semi-definit: $\mathbf{A} \geq 0,$ wenn $\mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{x} \geq 0 \quad \forall \mathbf{x} \neq \mathbf{0}$ falls $\mathbf{A}$ symmetrisch

$$\mathbf{A} = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} \in \mathbb{R}^{m \times n}$$

- Normen
  - l_p-Norm eines Vektors: $\|\mathbf{x}\|_p = \sqrt[p]{\sum_{i=1}^{m} |x_i|^p}$
  - l_p-induzierte Norm einer Matrix: $\|\mathbf{X}\|_p = \sqrt[p]{\sum_{i=1}^{m}\sum_{j=1}^{n} |a_{ij}|^p}$

Wiederholung: Eigenvektoren

- Gilt für Matrix $\mathbf{A}$, Vektor $\mathbf{v}$, und ein Skalar λ die Beziehung $\mathbf{Av} = \lambda\mathbf{v}$, dann heißen:
  - $\mathbf{v} \neq \mathbf{0}$ Eigenvektor und
  - $\lambda \in \mathbb{C}$ Eigenwert der Matrix.

- Symmetrische Matrix $\mathbf{A} \in \mathbb{R}^{m \times m}$:
  - Es gibt k Eigenwerte mit jeweils Vielfachheit l_k, so dass $\sum_{i=1}^{k} l_i = m$. Eigenvektoren bilden Unterraum der Dimensionen l_k.
  - Alle Eigenwerte sind reell
  - Spur $tr(\mathbf{A}) = \sum_{i=1}^{m} a_{ii} = \sum_{i=1}^{m} \lambda_i$ ist Summe aller Eigenwerte.

Wiederholung: Eigenvektoren

- Eigenwertzerlegung (für symmetrische Matrix $\mathbf{A}$),

$$\mathbf{A} = \mathbf{V}\Delta\mathbf{V}^{\mathrm{T}} = [\mathbf{v}_1 \quad \cdots \quad \mathbf{v}_m] \begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_m \end{bmatrix} [\mathbf{v}_1 \quad \cdots \quad \mathbf{v}_m]^{\mathrm{T}} \qquad \mathbf{v}_i^{\mathrm{T}}\mathbf{v}_j = \begin{cases} 1 & \text{falls } i = j \\ 0 & \text{falls } i \neq j \end{cases}$$

Eigenvektoren

Eigenwerte
(eindeutig, bis auf Permutation)

Orthonormalbasis

PCA
Bestimmen der Hauptkomponenten

- Gegeben Datenmatrix $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ mit m Zeilen
- Gesucht ist Matrix $\mathbf{A} = (\mathbf{a}_1, \ldots, \mathbf{a}_m)$, so dass:
  - Spalten $\mathbf{a}_1, \ldots, \mathbf{a}_m$ bilden Orthonormalbasis.
  - Spalte $\mathbf{a}_1$ erklärt möglichst viel Varianz der Daten.
  - Spalte $\mathbf{a}_2$ erklärt möglichst viel Restvarianz der Daten.
  - …
- Annahmen im Folgenden:
  - $\sum_{i=1}^{n} \mathbf{x}_i = \mathbf{0}$ (sonst: $\sum_{i=1}^{n} \mathbf{x}_i \,/\, n$ von jeder Spalte abziehen)
  - Sei $\mathbf{C} = \frac{1}{n-1} \mathbf{X}\mathbf{X}^{\mathrm{T}} \in \mathbb{R}^{m \times m}$ die Kovarianz der Daten:
    - Eigenwerte $\lambda_1, \ldots, \lambda_m > 0$ von $\mathbf{C}$ seien paarweise verschieden.

PCA
Bestimmen der ersten Hauptkomponente

- Wähle erste Hauptkomponente $\mathbf{a}_1$, so dass:
  - $\mathbf{a}_1^{\mathrm{T}}\mathbf{a}_1 = 1$ (Orthonormalbasis)
  - $\mathbf{a}_1^{\mathrm{T}}\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{a}_1 = \mathbf{a}_1^{\mathrm{T}}\mathbf{C}\mathbf{a}_1$ ist maximal (Varianz der durch $\mathbf{a}_1$ transformierten Daten)

PCA
Bestimmen der ersten Hauptkomponente

- Löse $\mathbf{a}_1 = \max_{\mathbf{a},\mathbf{a}^T\mathbf{a}=1} \mathbf{a}^T\mathbf{C}\mathbf{a}$
  - Suche Extremstellen $\mathbf{a}_1$ der Lagrangefunktion $L(\mathbf{a},\lambda)$: $L(\mathbf{a},\lambda) = \mathbf{a}^T\mathbf{C}\mathbf{a} - \lambda(\mathbf{a}^T\mathbf{a} - 1)$
  - Ableiten & Null setzen gibt: $\mathbf{C}\mathbf{a}_1 = \lambda_1\mathbf{a}_1$.
    - $\mathbf{a}_1$ ist Eigenvektor der Kovarianzmatrix $\mathbf{C}$.
  - Für die Kovarianz im Zielsystem (zu maximieren) gilt: $\mathbf{a}_1^T\mathbf{C}\mathbf{a}_1 = \mathbf{a}_1^T(\mathbf{C}\mathbf{a}_1) = \mathbf{a}_1^T(\lambda_1\mathbf{a}_1) = \lambda_1\mathbf{a}_1^T\mathbf{a}_1 = \lambda_1$
    - Wähle daher Eigenvektor mit größtem Eigenwert.

PCA
Bestimmen der zweiten Hauptkomponente

- Wähle zweite Hauptkomponente $\mathbf{a}_2$, so dass:
  - $\mathbf{a}_1^{\mathrm{T}}\mathbf{a}_1 = 1$ und
  - $\mathbf{a}_1^{\mathrm{T}}\mathbf{a}_2 = 0$ (Orthonormalbasis)
  - $\mathbf{a}_2^{\mathrm{T}}\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{a}_2 = \mathbf{a}_2^{\mathrm{T}}\mathbf{C}\mathbf{a}_2$ ist maximal (Varianz der durch $\mathbf{a}_2$ transformierten Daten)

PCA
Bestimmen der zweiten Hauptkomponente

- Löse $\mathbf{a}_2 = \max_{\substack{\mathbf{a},\mathbf{a}^T\mathbf{a}=1 \\ \mathbf{a}^T\mathbf{a}_1=0}} \mathbf{a}^T\mathbf{C}\mathbf{a}$
  - Suche Extremstellen $\mathbf{a}_2$ der Lagrangefunktion $L(\mathbf{a},\lambda,\delta)$:
    $L(\mathbf{a},\lambda,\delta) = \mathbf{a}^T\mathbf{C}\mathbf{a} - \lambda(\mathbf{a}^T\mathbf{a} - 1) - \delta\mathbf{a}^T\mathbf{a}_1$
  - Ableiten & Null setzen gibt: $\mathbf{C}\mathbf{a}_2 = \lambda_2\mathbf{a}_2$,
    - $\mathbf{a}_2$ ist Eigenvektor der Kovarianzmatrix $\mathbf{C}$.
  - Wieder gilt $\mathbf{a}_2^T\mathbf{C}\mathbf{a}_2 = \lambda_2$ (zu maximieren) und, da Eigenvektoren orthogonal sein sollen $\lambda_1 \neq \lambda_2$.
    - Wähle daher Eigenvektor mit zweitgrößtem Eigenwert.

PCA
Bestimmen der Hauptkomponenten

- Sind die Eigenwerte $\lambda_1 > \ldots > \lambda_m > 0$ paarweise verschieden, wähle für die i-te Hauptkomponente einen Eigenvektor mit Eigenwert λ_i.

- Hinweis: Sind zwei Eigenwerte $\lambda_i = \lambda_{i+1}$ identisch, wähle zueinander orthogonale Eigenvektoren mit entsprechendem Eigenwert als i- bzw. $(i+1)$-te Hauptkomponente

PCA

- Teilprobleme:
  - Bestimmen der Hauptkomponenten.
    - Hauptkomponenten entsprechen den Eigenvektoren (geordnet nach Eigenwert) der Kovarianzmatrix.
  - Transformation und Reduktion der Daten.
    - Transformation in das neue Koordinatensystem.
    - Welche Dimensionen können ignoriert werden?
  - Interpretierbarkeit der neuen Daten.
    - Wie kann Interpretierbarkeit der neuen Daten verbessert werden?

PCA
Transformation & Reduktion

- Transformiere Daten $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ durch gegebene Hauptkomponenten $\mathbf{A} = (\mathbf{a}_1, \ldots, \mathbf{a}_m)$ in neue Daten $\mathbf{Z} = \mathbf{A}^{\mathrm{T}} \cdot \mathbf{X}$.
  - Daten besitzen gleiche Dimension (m) wie zuvor.

- Aber es gilt (siehe Eigenwertzerlegung):
  - Kovarianz $\mathbf{C}^{\mathrm{T}} = \mathbf{A} \cdot \mathbf{\Lambda} \cdot \mathbf{A}^{\mathrm{T}}$.

  - Daher ändert sich die Summe der Varianzen

    $$tr(\mathbf{C}) = tr\left(\mathbf{A} \cdot \mathbf{\Lambda} \cdot \mathbf{A}^{\mathrm{T}}\right) = tr\left(\mathbf{\Lambda} \cdot \mathbf{A} \cdot \mathbf{A}^{\mathrm{T}}\right) = tr(\mathbf{\Lambda}) = \sum_{i=1}^{m} \lambda_i$$

    in den einzelnen Komponenten nicht.

PCA
Transformation & Reduktion

- Die Eigenvektoren mit den k größten Eigenwerten decken c Prozent der Gesamtvarianz ab, wobei:

$$c = \sum_{i=1}^{k} \lambda_i \Big/ tr(\mathbf{C})$$

- Wähle für Transformation nicht alle Hauptkomponenten sondern k Eigenvektoren mit den größten Eigenwerten $\mathbf{A}_{(k)} = (\mathbf{a}_1, \ldots, \mathbf{a}_k)$ und transformiere $\mathbf{Z} = \mathbf{A}_{(k)}^{\mathrm{T}} \cdot \mathbf{X}$
  - Zielraum nun k-dimensional (nicht m-dimensional)
  - Wähle k so, dass Abdeckung der Varianz dennoch ausreichend groß.

PCA
Algorithmus

- Seien Eigenwerte $\lambda_1, \ldots, \lambda_m > 0$ verschieden, $\sum_{i=1}^{n} \mathbf{x}_i = \mathbf{0}$
- Input: Daten $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$, minimale Abdeckung c
- Setze $k=0$, $\mathbf{C} = \frac{1}{n-1} \mathbf{X}\mathbf{X}^{\mathrm{T}}$
- Wiederhole
  - Setze $k=k+1$.
  - Bestimme Eigenvektor $\mathbf{a}_k$ von $\mathbf{C}$ mit dem k größtem Eigenwert.
- Bis $c < \sum_{i=1}^{k} \lambda_i \Big/ tr(\mathbf{C})$.
- Transformiere Daten $\mathbf{Z} = (\mathbf{a}_1, \ldots, \mathbf{a}_k)^{\mathrm{T}} \cdot \mathbf{X}$

PCA
Nachteile

- Problematisch bei schlecht skalierten Daten.
  - Informationsreiche Komponenten mit geringer Varianz werden möglicherweise entfernt.
- Kovarianzmatrix ist quadratisch in Anzahl der Attribute:
  - Bei Texten mit 100.000 verschiedenen Wörtern hat diese 10 Mrd. Einträge (8 Byte pro Eintrag: 80Gb).
  - Idee: $\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{a} = \lambda\mathbf{a}$

    $\Rightarrow \mathbf{X}\cdot\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{a} = \mathbf{X}\cdot\lambda\mathbf{a} \Rightarrow \mathbf{X}\mathbf{X}^{\mathrm{T}}\cdot(\mathbf{X}\mathbf{a}) = \lambda(\mathbf{X}\mathbf{a})$

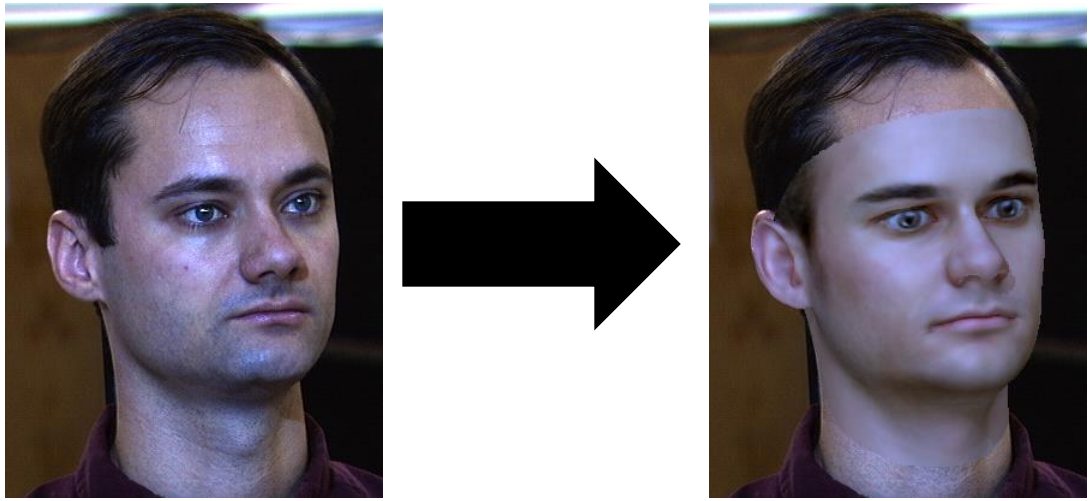
    - Berechne Eigenvektoren $\mathbf{a}$ von $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ (n x n-Matrix).
    - Vektoren $\mathbf{X}\mathbf{a}$ sind gesuchte Eigenvektoren von $\mathbf{X}\mathbf{X}^{\mathrm{T}}$ mit gleichem Eigenwert.

PCA

- Teilprobleme:
  - Bestimmen der Hauptkomponenten.
    - Hauptkomponenten entsprechen den Eigenvektoren (geordnet nach Eigenwert) der Kovarianzmatrix.
  - Transformation und Reduktion der Daten.
    - Transformation durch die Eigenvektoren mit den k höchsten Eigenwerten.
  - Interpretierbarkeit der neuen Daten.
    - Wie kann Interpretierbarkeit der neuen Daten verbessert werden?

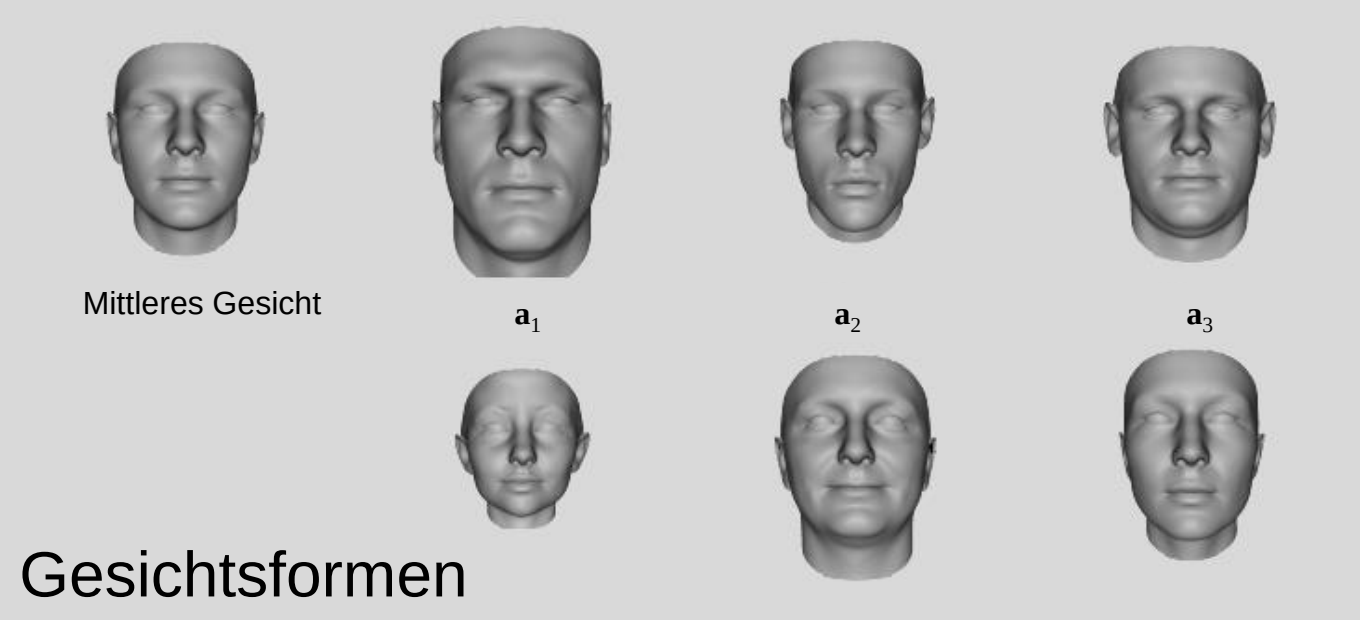
PCA
Anwendung & Interpretation

- Anwendungsbeispiel:
  - Morphace (Universität Basel)
    - 3D-Modelle von 200 verschiedenen Personen (jeweils über 150000 Feature)
    - PCA mit 199 Hauptkomponenten, jedes (3D) Gesicht wird durch 199 Parameter charakterisiert.



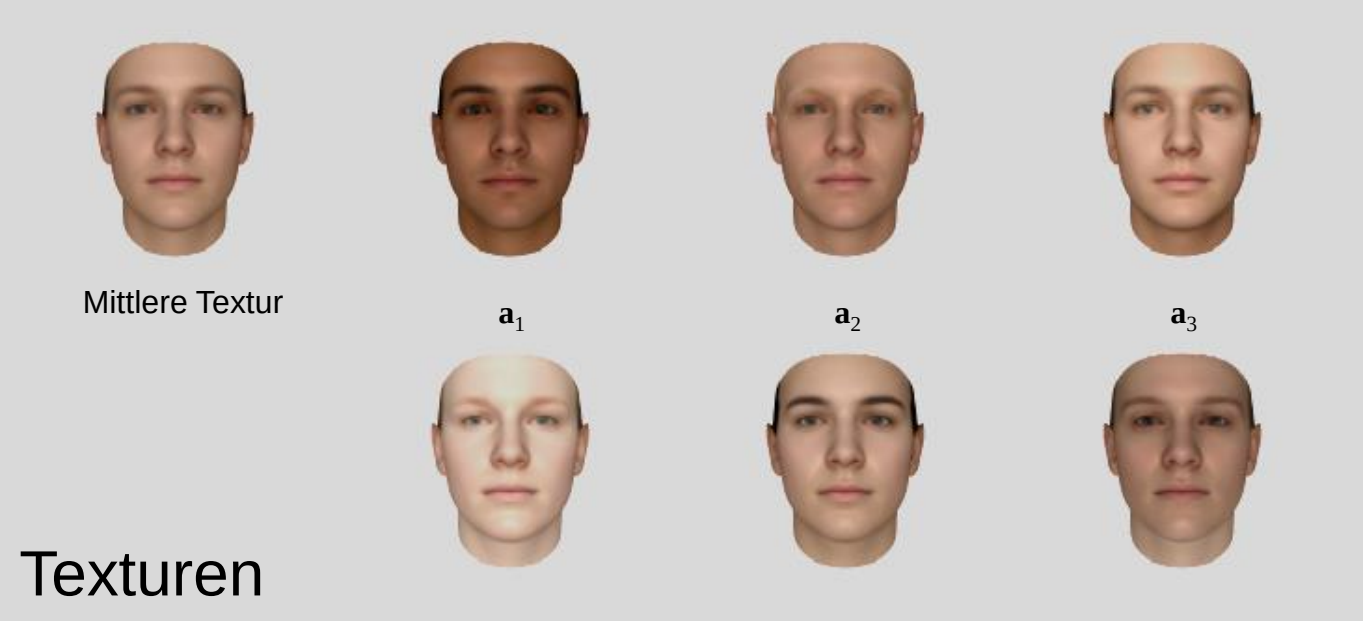
PCA
Anwendung & Interpretation

- Anwendungsbeispiel:
  - Morphace (Universität Basel)
    - Visualisierung der Hauptkomponenten im Originalraum



PCA
Anwendung & Interpretation

- Anwendungsbeispiel:
  - Morphace (Universität Basel)
    - Visualisierung der Hauptkomponenten im Originalraum



PCA
Anwendung & Interpretation

- Anwendungsbeispiel:
  - PCA bei Texten.
  - Hauptkomponenten als Linearkombination aller möglichen Worte:
    - Bsp: 0.001*{season}+0.03*{million}-0.01*{school}+…
  - Schwer zu interpretieren.
  - Besser sparse Hauptkomponenten:

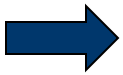
1st PC	2nd PC	3rd PC	4th PC	5th PC
Million	Point	Official	president	School
Percent	Play	Government	Campaign	Program
Business	Team	United_States	Bush	Children
Company	Season	U_S	Administration	Student
Market	Game	attack		
Companies				

5 sparse Hauptkomponenten von Artikeln der NYTimes

Sparse PCA

- 1. Möglichkeit:
  - Schritt 1: Hauptkomponenten über Standard-PCA berechnen
  - Schritt 2: Hauptkomponenten in sparse Vektoren überführen, durch Ersetzen von kleinen Werten mit 0.
- Beispiel:

	σ(X)	1st PC	2nd PC
X_1	75.75	0.956	-0.288
X_2	13.13	0.294	0.945
X_3	0.61	0.015	-0.154
X_4	0.02	0.001	-0.002
λ		82.308	6.739



1st SPC	2nd SPC
1	0
0	1
0	0
0	0

Sparse PCA

- 1. Möglichkeit (Probleme):
  - Auf Orthogonalität muss explizit geachtet werden.
  - Korrelationen zwischen Variablen im Originalraum und Zielraum werden ignoriert:

	σ(X)	1st PC	2nd PC
X_1	75.75	0.956	-0.288
X_2	13.13	0.294	0.945
X_3	0.61	0.015	-0.154
X_4	0.02	0.001	-0.002
λ		82.308	6.739

$$\rho(Z_1, X_2) = \frac{Cov(Z_1, X_2)}{\sqrt{Var(Z_1)Var(X_2)}} = \frac{\lambda a_{12}}{\sqrt{\lambda} \cdot \sigma(x_2)}$$

$$= 0.736$$

$$\rho(Z_2, X_2) = 0.677$$

  - 2. Dimension des Originalraums korreliert stärker mit der 1. Dimension des Zielraums (1. Eigenvektor) als mit der 2. Dimension des Zielraums (2. Eigenvektor).

Sparse PCA

- 2. Möglichkeit:
  - Wähle Hauptkomponenten so, dass möglichst viele Einträge 0 sind.
  - Wähle beispielsweise $\mathbf{a}_1$ derart, dass:
    - $\mathbf{a}_1^{\mathrm{T}}\mathbf{C}\mathbf{a}_1$ maximal, unter den Bedingungen:
    - $\mathbf{a}_1^{\mathrm{T}}\mathbf{a}_1 = 1$ (normiert)
    - $\|\mathbf{a}_1\|_0 = \sum_{j=1}^{m} [\![a_{1j} > 0]\!] \leq t$ für ein wählbares t
  - Problem: $\|\mathbf{a}_1\|_0$ ist nicht stetig und daher schwer zu optimieren.

Sparse PCA

- 2. Möglichkeit (Relaxierung der l_0-Norm):
  - Wähle Hauptkomponenten so, dass möglichst viele Einträge 0 sind,
  - Wähle beispielsweise $\mathbf{a}_1$ derart, dass:
    - $\mathbf{a}_1^T \mathbf{C} \mathbf{a}_1$ maximal, unter den Bedingungen:
    - $\mathbf{a}_1^T \mathbf{a}_1 = 1$ (normiert)
    - $\|\mathbf{a}_1\|_1 = \sum_{j=1}^{m} |a_{1j}| \leq t$ für ein wählbares t
  - Problem: Kein konvexes Optimierungsproblem, Lösungen sind im Allgemeinen nur lokal-optimal.

Sparse PCA

- 3. Möglichkeit:
  - Nehme Kardinalität der Hauptkomponenten in Optimierungsfunktion auf.
  - Wähle beispielsweise Hauptkomponente $\mathbf{a}_1$ so, dass:
    - $\mathbf{a}_1^T \mathbf{C} \mathbf{a}_1 - \rho \|\mathbf{a}_1\|_0$ maximal, unter den Bedingungen:
    - $\mathbf{a}_1^T \mathbf{a}_1 = 1$ (normiert)
  - Problem: Kein konvexes und nicht stetiges Optimierungsproblem.
  - Idee: Finde konvexe obere Schranke der Funktion und optimiere diese.

Sparse PCA

- 3. Möglichkeit:
  - Umformulieren & Relaxieren des Optimierungsproblems:

$$\max_{\mathbf{a},\|\mathbf{a}\|_2=1} \mathbf{a}^T\mathbf{C}\mathbf{a} - \rho\|\mathbf{a}\|_0 = \max_{\substack{\mathbf{A},\mathbf{A}\geq 0 \\ tr(\mathbf{A})=1 \\ rk(\mathbf{A})=1}} tr(\mathbf{AC}) - \rho\sqrt{\|\mathbf{A}\|_0}$$

Nutze $\mathbf{A} = \mathbf{a}\mathbf{a}^T$

$$\leq \max_{\substack{\mathbf{A},\mathbf{A}\geq 0 \\ tr(\mathbf{A})=1 \\ rk(\mathbf{A})=1}} tr(\mathbf{AC}) - \rho\|\mathbf{A}\|_1/\|\mathbf{A}\|_2$$

Beziehung Normen

$$\leq \max_{\substack{\mathbf{A},\mathbf{A}\geq 0 \\ tr(\mathbf{A})=1}} tr(\mathbf{AC}) - \rho\|\mathbf{A}\|_1$$

L_2–Norm ist hier 1
Ignoriere Rang

  - Rang der Lösung kann >1 sein:
    - Approximiere durch $\mathbf{a}\mathbf{a}^T$, wobei $\mathbf{a}$ der Eigenvektor mit größtem Eigenwert ist.

Sparse PCA

- 3. Möglichkeit:
  - Optimierungsproblem $\max_{\mathbf{A},\mathbf{A}\geq 0,\ tr(\mathbf{A})=1} tr(\mathbf{AC}) - \rho\|\mathbf{A}\|_1$ ist konvex.
  - Eine Optimierung mit Hilfe eines Koordinatenaufstiegs ist möglich.
    - Laufzeit in diesem Falle im Bereich $O(m^3)$; dies ist für ein großes Vokabular zu groß.

Sparse PCA

- 3. Möglichkeit:
  - Optimierungsproblem $\max_{\mathbf{A},\mathbf{A}\geq 0,\ tr(\mathbf{A})=1} tr(\mathbf{AC}) - \rho\|\mathbf{A}\|_1$ ist konvex.
  - Idee: Entferne Wörter mit geringer Varianz zur Verbesserung der Laufzeit:
    - Wörter mit geringer Varianz werden niemals Teil der nächsten sparsen Hauptkomponente.
    - Entferne Wort i, wenn $\mathbf{C}_{ii} \leq \rho$
      - Sicheres Entfernen, d.h. es wird mit Sicherheit kein Wort zu viel entfernt (Beweis auf folgenden Folien)
      - In Experimenten beispielsweise ~500 statt ~100.000 Wörter, wenn pro Komponente ca. 5 Wörter angestrebt sind
        - Parameter ρ wird passend eingestellt.

Sparse PCA

- Beweis:
  - 1.: definiere $\alpha(\mathbf{a}) = \mathbf{a}^{\mathrm{T}}\mathbf{C}\mathbf{a} - \rho\|\mathbf{a}\|_0$
  - gesucht: $\max_{\mathbf{a},\mathbf{a}^{\mathrm{T}}\mathbf{a}=1} \alpha(\mathbf{a}) = \max_{\mathbf{a},\mathbf{a}^{\mathrm{T}}\mathbf{a}\leq 1} \alpha(\mathbf{a})$ Konvexe Funktion, daher Maxima an Rändern
  - 2.: setze $\mathbf{a} = \mathbf{D}_{\mathbf{a}}\mathbf{y}_{\mathbf{a}}$ für binäre Diagonalmatrix $\mathbf{D}_{\mathbf{a}} \in \{0,1\}^{m\times m}$ und Vektor $\mathbf{y}_{\mathbf{a}} \in \mathbb{R}^m, \mathbf{y}_{\mathbf{a}}^{\mathrm{T}}\mathbf{y}_{\mathbf{a}} = 1$ (nicht eindeutig)
    - Dann gilt für $\beta(\mathbf{D},\mathbf{y}) = \mathbf{y}^{\mathrm{T}}\mathbf{D}\mathbf{C}\mathbf{D}\mathbf{y} - \rho\mathbf{1}^{\mathrm{T}}\mathbf{D}\mathbf{1}$:

      $\beta(\mathbf{D}_{\mathbf{a}},\mathbf{y}_{\mathbf{a}}) \leq \alpha(\mathbf{a})$, wobei für die Gleichheit gilt:

      $d_{ii} = a_i > 0 \;, \mathbf{y} = \mathbf{a} \Rightarrow \beta(\mathbf{D}_{\mathbf{a}},\mathbf{y}_{\mathbf{a}}) = \alpha(\mathbf{a})$

Sparse PCA

- Beweis:
  - 3.: gesucht ist daher (Alternative)

$$\max_{\mathbf{D}\in\{0,1\}^{m\times m}_{diag}} \max_{\mathbf{y},\mathbf{y}^{\mathrm{T}}\mathbf{y}=1} \mathbf{y}^{\mathrm{T}}\mathbf{DCDy} - \rho\mathbf{1}^{\mathrm{T}}\mathbf{D1}$$

$$= \max_{\mathbf{D}\in\{0,1\}^{m\times m}_{diag}} \max_{\mathbf{y},\mathbf{y}^{\mathrm{T}}\mathbf{y}=1} \frac{1}{n-1}\mathbf{y}^{\mathrm{T}}\mathbf{DX}^{\mathrm{T}}\mathbf{XDy} - \rho\mathbf{1}^{\mathrm{T}}\mathbf{D1}$$

Definition Kovarianz

$$= \max_{\mathbf{D}\in\{0,1\}^{m\times m}_{diag}} \max_{\mathbf{y},\mathbf{y}^{\mathrm{T}}\mathbf{y}=1} \frac{1}{n-1}\mathbf{y}^{\mathrm{T}}\mathbf{XDX}^{\mathrm{T}}\mathbf{y} - \rho\mathbf{1}^{\mathrm{T}}\mathbf{D1}$$

Vertausche X und D
(hier möglich!)

$$= \max_{\mathbf{D}\in\{0,1\}^{m\times m}_{diag}} \max_{\mathbf{y},\mathbf{y}^{\mathrm{T}}\mathbf{y}=1} \frac{1}{n-1}\mathbf{y}^{\mathrm{T}}\left(\sum_{i=1}^{m} d_{ii}\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}\right)\mathbf{y} - \rho\mathbf{1}^{\mathrm{T}}\mathbf{D1}$$

Produkt zerlegen,
$\mathbf{x}_i$ ist i-te Zeile/Feature von $\mathbf{X}$

Sparse PCA

- Beweis:
  - 3.: gesucht ist daher (Fortsetzung):

$$\max_{\mathbf{D}\in\{0,1\}^{m\times m}_{diag}} \max_{\mathbf{y},\mathbf{y}^{\mathrm{T}}\mathbf{y}=1} \frac{1}{n-1}\mathbf{y}^{\mathrm{T}}\left(\sum_{i=1}^{m} d_{ii}\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}\right)\mathbf{y} - \rho\mathbf{1}^{\mathrm{T}}\mathbf{D}\mathbf{1}$$

$$= \max_{\mathbf{D}\in\{0,1\}^{m\times m}_{diag}} \max_{\mathbf{y},\mathbf{y}^{\mathrm{T}}\mathbf{y}=1} \sum_{i=1}^{m}\left[\frac{1}{n-1} d_{ii}\left(\mathbf{x}_i^{\mathrm{T}}\mathbf{y}\right)^2 - \rho d_{ii}\right]$$

Ausklammern

$$= \max_{\mathbf{y},\mathbf{y}^{\mathrm{T}}\mathbf{y}=1} \sum_{i=1}^{m} \max_{d_{ii}\in\{0,1\}}\left[\frac{1}{n-1} d_{ii}\left(\mathbf{x}_i^{\mathrm{T}}\mathbf{y}\right)^2 - \rho d_{ii}\right]$$

Maximum in Summanden ziehen

$$= \max_{\mathbf{y},\mathbf{y}^{\mathrm{T}}\mathbf{y}=1} \sum_{i=1}^{m} \max\left[\frac{1}{n-1}\left(\mathbf{x}_i^{\mathrm{T}}\mathbf{y}\right)^2 - \rho, 0\right]$$

Einträge sind 0 oder 1

Sparse PCA

- Beweis:
  - 4.: Ein Summand in der Gleichung

$$\max_{\mathbf{y},\mathbf{y}^{\mathrm{T}}\mathbf{y}=1} \sum_{i=1}^{m} \max\left[\frac{1}{n-1}\left(\mathbf{x}_i^{\mathrm{T}}\mathbf{y}\right)^2 - \rho, 0\right]$$

ist immer 0 (unabhängig von $\mathbf{y}$) wenn gilt:

$$\frac{1}{n-1}\left(\mathbf{x}_i^{\mathrm{T}}\mathbf{x}_i\right)^2 = \mathbf{C}_{ii} < \rho$$

In diesem Fall ist $d_{ii} = 0$ und daher $a_i = 0$

PCA

- Teilprobleme:
  - Bestimmen der Hauptkomponenten.
    - Hauptkomponenten entsprechen den Eigenvektoren (geordnet nach Eigenwert) der Kovarianzmatrix.
  - Transformation und Reduktion der Daten.
    - Transformation durch die Eigenvektoren mit den k höchsten Eigenwerten.
  - Interpretierbarkeit der neuen Daten.
    - Sparse Hauptkomponenten erhöhen die Interpretierbarkeit der transformierten Daten.

Zusammenfassung
PCA

- PCA (Hauptkomponentenanalyse) projeziert Daten in neuen Raum:
  - Alle Komponenten sind unkorreliert.
  - Die Gesamtvarianz bleibt erhalten.
  - Die i-te Komponente hat größere Varianz als die $(i+1)$-te:
    - Ermöglicht das Weglassen hinterer Komponenten (Reduzierung der Dimension) ohne Varianz in den Daten stark zu beeinträchtigen.

Zusammenfassung
Sparse PCA

- PCA (Hauptkomponentenanalyse) projeziert Daten in neuen Raum.
- Sparse PCA erzeugt dabei sparse Hauptkomponenten.
  - Hauptkomponenten sind besser interpretierbar.
  - Sparse Daten sind auch im transformierten Raum sparse.
  - Wirkt regularisierend.
  - Ermöglicht sichere Reduzierung der Feature vor dem Berechnen der Hauptkomponenten und kann daher sehr viel schneller sein als PCA.
    - Im Worst-Case ist PCA laufzeittechnisch jedoch besser als Sparse PCA (quadratisch statt kubisch)

Fragen?