

Mustererkennung

Merkmalsreduktion

R. Neubecker, WS 2016 / 2017

Übersicht

2

Merkmale

Principal Component Analysis:

Konzept

Mathematische Formulierung

Anwendung

Eigengesichter

Merkmale

3

(Zu) Viele Merkmale

- Nicht immer kann ein BV-System problemspezifisch so programmiert werden, dass optimale / problemangepasste Merkmale verfügbar sind
- Universell einsetzbare BV-Bibliotheken liefern (viele) Standard-Merkmale
- Beispiel Matlab-Funktion regionprops():

Shape Measurements: 'Area', 'EulerNumber', 'Orientation', 'BoundingBox', 'Extent', 'Perimeter', 'Centroid', 'Extrema', 'PixelIdxList', 'ConvexArea', 'FilledArea', 'PixelList', 'ConvexHull', 'FilledImage', 'Solidity', 'ConvexImage', 'Image', 'SubarrayIdx', 'Eccentricity', 'MajorAxisLength', 'EquivalentDiameter', 'MinorAxisLength'.
Pixel Value Measurements: 'MaxIntensity', 'MinIntensity', 'WeightedCentroid', 'MeanIntensity', 'PixelValues'

- Beispiel Halcon, nur für OCR-Anwendungen:

'anisometry', 'chord_histo', 'compactness', 'convexity', 'cooc', 'foreground', 'foreground_grid_16', 'foreground_grid_9', 'gradient_8dir', 'height', 'moments_centra', 'moments_gray_plane', 'moments_region_2nd_invar', 'moments_region_2nd_rel_invar', 'moments_region_3rd_invar', 'num_connect', 'num_holes', 'num_runs', 'phi', 'pixel', 'pixel_binary', 'pixel_invar', 'projection_horizontal', 'projection_horizontal_invar', 'projection_vertical', 'projection_vertical_invar', 'ratio', 'width', 'zoom_factor'

Merkmale: Normierung

4

Skalen

- Werte verschiedener Merkmale können sehr unterschiedlich sein
→ untereinander nicht gut vergleichbar / kombinierbar
- Mögliche Abhilfe: auf ähnliche Wertebereiche bringen
→ Mittelwert $\mu = 0$, Standardabweichung $\sigma = 1$
- Gegeben: Stichprobe von N Merkmalsvektoren \vec{x}_n , $n = 1 \dots N$
Jedes Merkmal = jede Komponente $\vec{x}_n = (x_{1n}, x_{2n}, \dots, x_{in})$ skalieren:

$$\mu_i = \frac{1}{N} \sum_n x_{in}, \quad \sigma_i = \sqrt{\frac{1}{N-1} \sum_n (x_{in} - \mu_i)^2}$$

$$\hat{x}_{in} = \frac{x_{in} - \mu_i}{\sigma_i}$$

(aber: Werte liegen prinzipiell immer noch $-\infty < \hat{x}_{in} < \infty$)

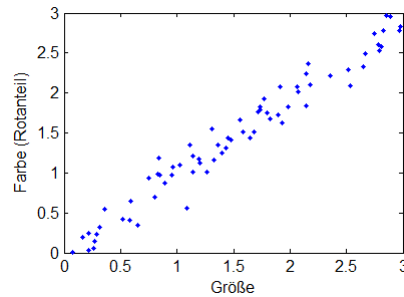
- Alternative: Skalierung auf Intervall $[\min(x_{in}) \max(x_{in})] \rightarrow [0 \ 1]$.
Aber: nächste Stichprobe kann anderes Intervall haben
→ normierte Werte werden außerhalb von $[0 \ 1]$ liegen

Merkmale: Korrelationen

5

Zu viele Merkmale

- Klassifikationsaufwand (inkl. Training) steigt mit Anzahl der Merkmale (=Dimension des Merkmalsraums)
- Wozu unnötige Merkmale mitschleppen?
Idee: nur unkorrelierte Merkmale benutzen, mit maximalem "Informationsgehalt"
- Korrelierte Merkmale
 - Beispiel: Merkmale *Größe* und *Farbe* (=Rotanteil) bei Äpfeln & Birnen
→ korreliert, weil Früchte wachsen und dabei an Grün verlieren
 - Redundanter Informationsgehalt, eigentlich ist eines der Merkmale ist ausreichend, aber:
Wie würde "optimales" Merkmal liegen?



Übersicht

6

Merkmale

Principal Component Analysis:

Konzept

Mathematische Formulierung

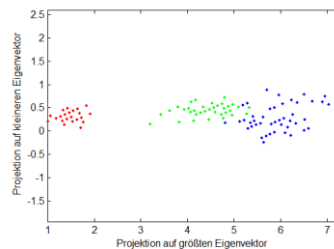
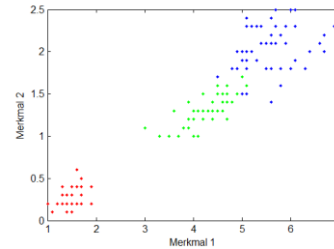
Anwendung

Eigengesichter

Principal Component Analysis (PCA)

Anschaulich

- Beispiel: Datensatz ist in beiden Merkmalen gleichermaßen gut klassifizierbar. Gleichzeitige Verwendung beider Merkmale gleichzeitig bringt keinen Vorteil, bedeutet nur Rechenaufwand bei Klassifikation
- PCA =
 - Projektion auf die Diagonale (und die Richtung senkrecht dazu) = Drehung des Koordinatensystems
 - Vernachlässigung des zweiten neuen Merkmals (senkrechte Achse in 2. Plot) → wenig Informationsverlust



Principal Component Analysis (PCA)

- Auch bekannt als "Hauptkomponentenanalyse" oder "Karhunen-Loève-Transformation"
- Verfahren zur Datenreduktion / -kompression, auch nutzbar zur Visualisierung hochdimensionaler Daten
- Nichts anderes als eine lineare Hauptachsentransformation des Koordinatensystems unseres Merkmalsraums + Auswahl der "wichtigsten" Achsen der neuen Koordinaten
- Im Beispiel: Projektion 2-dimensionale Merkmale auf eine Gerade (1-dim.). Allgemein: Projektion des D -dimensionalen Merkmalsraums auf einen M -dimensionalen Unterraum, $M < D$
- Mögliche Forderungen (führen zum gleichen Verfahren):
 - Informationsgehalt der projizierten Daten möglichst hoch (Maximum Variance)
 - Abweichung von Originaldaten soll minimal sein (Minimum Error)
- Ausgangspunkt: Stichprobe von N Merkmalsvektoren \vec{x}_n , alle Merkmale sind bereits zentriert (Mittelwerte = 0 → dann reicht Drehung aus)

Merkmale

Principal Component Analysis:

Konzept

Mathematische Formulierung

Anwendung

Eigengesichter

Principal Component Analysis: Formulierung

PCA-Formulierung: Maximale Varianz (1)

- Forderung: Varianz ("Informationsgehalt") der projizierten Daten soll maximal sein
- Für den Anfang: Projektion auf 1 Dimension $D = 2 \rightarrow M = 1$
(= auf eine Gerade \leftarrow beschrieben durch einen Vektor in Richtung der Geraden)
- Projektion eines Merkmalsvektors auf Richtung eines Einheitsvektors \vec{u}

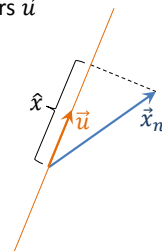
$$\vec{x}_n \rightarrow \hat{x}_n = \vec{u}^t \cdot \vec{x}_n$$

- Varianz der projizierten Merkmalsvektoren

$$\sigma^2 = \frac{1}{N-1} \sum_n (\hat{x}_n - \hat{\mu})^2 = \dots = \vec{u}^t \vec{S} \vec{u}$$

mit Mittelwert $\hat{\mu} = \dots = \vec{u}^t \vec{\mu}$

und Kovarianzmatrix $\vec{S} = \frac{1}{N-1} \sum_n (\vec{x}_n - \vec{\mu})(\vec{x}_n - \vec{\mu})^t$



PCA-Formulierung: Maximale Varianz (2)

- Gesucht: \vec{u}
- Forderung: Maximierung der Varianz σ^2 , unter Nebenbedingung $|\vec{u}| = 1$

→ Lagrangefunktion

$$L = \vec{u}^t \vec{S} \vec{u} + \alpha (1 - \vec{u}^t \vec{u}) \rightarrow \max, \quad \alpha: \text{Lagrange-Multiplikator}$$

- Ist erfüllt für $\vec{\nabla}_{\vec{u}} L = 0 \rightarrow$

$$\vec{S} \vec{u} = \lambda \vec{u}$$

→ \vec{u} ist Eigenvektor der Kovarianzmatrix

- Gesuchte Varianz $\sigma^2 = \vec{u}^t \vec{S} \vec{u} = \lambda =$ zugehöriger Eigenwert

PCA-Formulierung: Maximale Varianz → Verallgemeinerung

- Verallgemeinerung = Projektion von \vec{x}_n auf einen M -dimensionalen Unterraum
- Allgemein gilt: Eigenvektoren sind orthogonal zueinander
→ spannen ein eigenes kartesisches Koordinatensystem auf
- Die Eigenvektoren \vec{u}_i der Kovarianzmatrix haben die besondere Eigenschaft, dass die zugehörigen Eigenwerte $\lambda_i = \sigma^2$ gleich der Varianz der auf diese Richtung projizierten (Merkmals-) Vektoren \vec{x}_n sind
- Forderung: maximale Varianz → Benutze die zu den größten Eigenwerten λ_i gehörende Eigenvektoren, um (Merkmals-) Vektoren mit weniger Dimensionen ($M < D$) anzunähern:

$$\vec{x}_n = \sum_{j=1}^D \alpha_{nj} \vec{u}_j = \sum_{j=1}^M \alpha_{nj} \vec{u}_j + \sum_{j=M+1}^D \alpha_{nj} \vec{u}_j \approx \tilde{\vec{x}}_n = \sum_{j=1}^M \alpha_{nj} \vec{u}_j$$

mit $\alpha_{nj} = \vec{x}_n^t \cdot \vec{u}_j$

- Bzw.: die zu den kleinsten Eigenwerten (λ_j mit $j > M$) gehörenden Eigenvektoren können ohne großen Informationsverluste weg gelassen werden

Alternative PCA-Formulierung: Minimaler Projektionsfehler

- Start mit orthonormaler Basis $\vec{u}_j^t \vec{u}_k = \delta_{jk}$
Jeder Vektor kann in dieser Basis dargestellt werden $\vec{x}_n = \sum_j \alpha_{nj} \vec{u}_j$ mit $\alpha_{nj} = \vec{x}_n^t \vec{u}_j$
- Beschränkung auf Unterraum $M < D$

$$\vec{x}_n = \sum_{j=1}^D \alpha_{nj} \vec{u}_j \approx \tilde{\vec{x}}_n = \sum_{j=1}^M \alpha_{nj} \vec{u}_j \quad (\text{s. o.})$$

- Abschätzung des Fehlers mit

$$J = \frac{1}{N} \sum_{n=1}^N |\vec{x}_n - \tilde{\vec{x}}_n|^2$$

- Gesucht: \vec{u}_i für minimalen Fehler $J \rightarrow \dots \rightarrow$ Lagrangefunktion wie oben. Ist erfüllt für

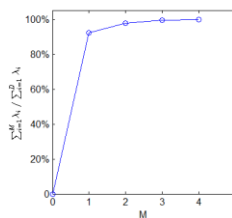
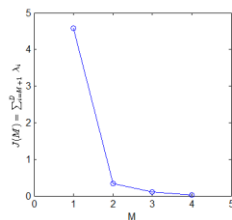
$$\vec{S} \vec{u}_j = \lambda_j \vec{u}_j$$

- Gesuchter Fehler: Summe der "nicht benutzten" Eigenwerte

$$J = \sum_{j=M+1}^D \lambda_j$$

(Verlustbehaftete) Datenreduktion

- Koordinaten \vec{u}_i des neuen Merkmalsraums nach "Informationsgehalt" sortiert, d.h. Weglassen der letzten Koordinaten bedeutet geringen Informationsverlust
- Quantitatives Maß = Fehler J



Beispiel: ursprünglich 4-dimensionaler Merkmalsraum, per PCA transformiert. In der 1. Koordinate sind bereits 92% der "Information" enthalten.

- Daraus ableitbar: wie viele der neuen Merkmalsdimensionen müssen mitgenommen werden

Zusammenfassung Konzept der PCA

- Stichprobe von (Merkmals-) Vektoren \vec{x}_n mit D Dimensionen soll (mit möglichst geringem Informationsverlust) durch Vektoren $\vec{\tilde{x}}_n$ mit weniger Dimensionen ($M < D$) dargestellt werden
- "Projektion auf M -dimensionalen Unterraum" :
 - a) Darstellung der Vektoren in gedrehtem Koordinatensystem
 - b) Weglassen einiger dieser Koordinaten
- Neues Koordinatensystem = Eigenvektoren \vec{u}_i der Kovarianzmatrix \vec{S} von \vec{x}_n , Zugehörige Eigenwerte λ_i enthalten Information über "Wichtigkeit" der jeweiligen Koordinate → Sortierung nach Größe der Eigenwerte
- Summe über die nicht benutzten Eigenwerte = Fehler durch weggelassene Koordinaten

Merkmale

Principal Component Analysis:

Konzept

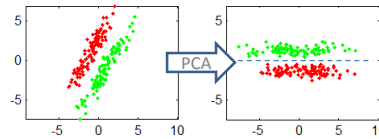
Mathematische Formulierung

Anwendung

Eigengesichter

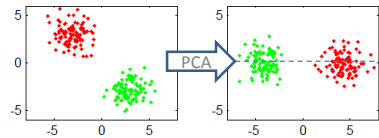
PCA & Klassifikation

- PCA reduziert nur die Merkmals-Datenmenge, ist per se keine Hilfe bei der Klassifikation
- Grundlage ist die Kovarianz der gesamten Stichprobe - ohne Einbeziehung der Klassenzugehörigkeit



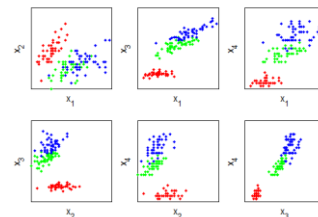
PCA vs. Fisher's Linearer Diskriminante

- Fisher benutzt den Abstand der Klassenmittelpunkte (Between-Class-) und die Streuung der Datenpunkte innerhalb jeder Klasse (Within-Class-Scattermatrix) – die beide zur Gesamtvarianz beitragen
- PCA benutzt nur die Gesamtvarianz, d.h. führt nur zu Separierbarkeit, falls Abstand der Klassenmittelpunkte \gg Streuung innerhalb der Klassen ist

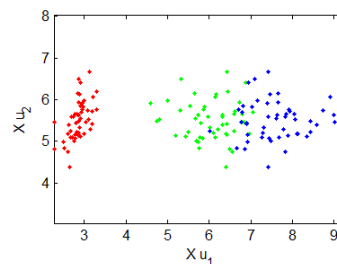


PCA zur Datenvisualisierung

- Visualisierung von Daten ist nur in 2 oder 3 Dimensionen möglich
- Wie vorgehen bei mehr Merkmalen / höherdimensionalem Merkmalsraum? Kombinatorisches Gegenüberstellen von je 2, bzw. je 3 Merkmalen ist nur begrenzt sinnvoll
- Visualisierung (für Klassifikation) sollte die wesentlichen Informationen (z.B. relative Abstände zwischen Datenpunkten) wiedergeben
- Projektion hochdimensionaler Daten auf 2 oder 3 Dimensionen? PCA!



PCA



Merkmale**Principal Component Analysis:**

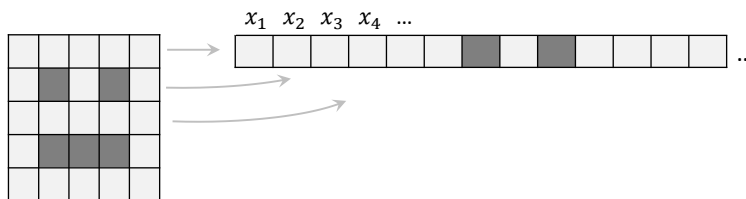
Konzept

Mathematische Formulierung

Anwendung

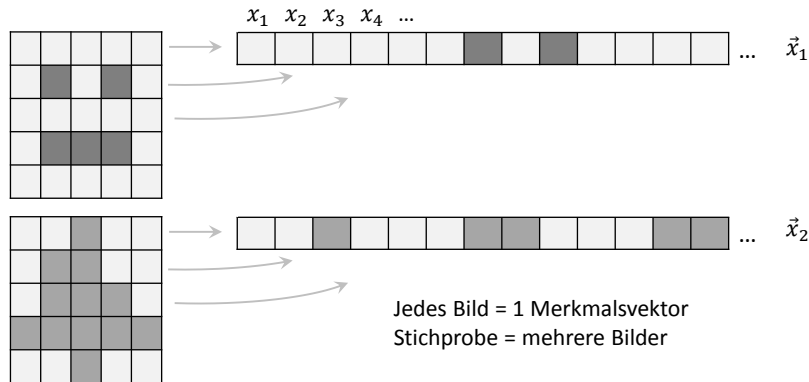
Eigengesichter**Principal Component Analysis: Eigengesichter****Triviale Bild-Merkmale**

- Extremes Beispiel für Wahl der Merkmale eines Bildes: Grauwerte jedes Pixels
→ Bild mit $n \times m$ Pixeln hat $n \cdot m$ Merkmale
(Merkmalsraum mit $n \cdot m$ Dimensionen)
- Anordnung der Pixel hier egal → räumliche Beziehungen spielen keine Rolle!
Pixel können auch linear hintereinander geschrieben werden



Triviale Bild-Merkmale

- Extremes Beispiel für Wahl der Merkmale eines Bildes: Grauwerte jedes Pixels
→ Bild mit $n \times m$ Pixeln hat $n \cdot m$ Merkmale
(Merkmalsraum mit $n \cdot m$ Dimensionen)
- Anordnung der Pixel hier egal → räumliche Beziehungen spielen keine Rolle!
Pixel können auch linear hintereinander geschrieben werden

**PCA-Anwendung: Eigenfaces**

- PCA auf Stichprobe von vielen Bildern von mehreren Gesichtern
→ Eigenvektoren \vec{u}_i = Grundkomponenten eines Gesichtes
("Eigen-Faces") = Basis"vektoren" zur Darstellung von Gesichtern
- Jedes Gesicht $\vec{X}_k = \sum_j \alpha_{kj} \vec{u}_j$ wird repräsentiert durch seine Koeffizienten $\alpha_{kj} = \vec{X}_k^t \cdot \vec{u}_j$ bezüglich dieser Basis
- Starke Datenkompression, pro Gesicht nicht mehr $n \cdot m$ Grauwerte, sondern nur noch die Koeffizienten für die wichtigsten Basisvektoren

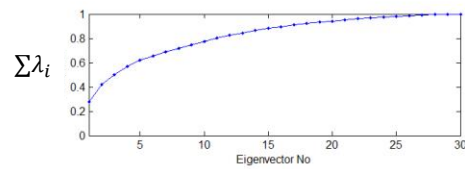
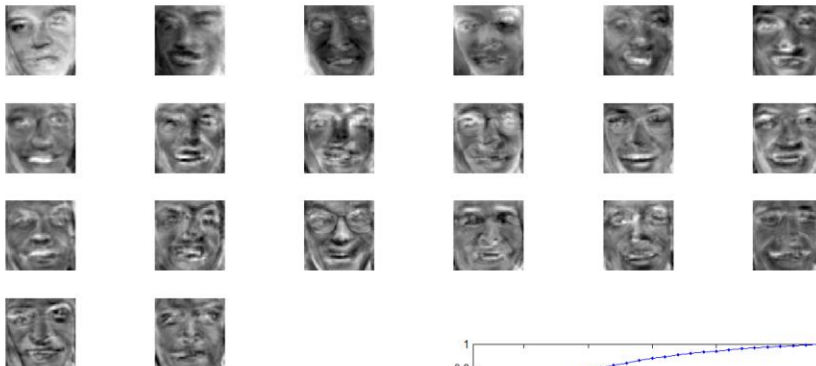
**Gesichter-Erkennung**

- Zerlegung eines unbekanntes Gesichtes \vec{Y} nach der Eigenfaces-Basis $\beta_j = \vec{Y}^t \vec{u}_j$
- Vergleich der gewonnenen Koeffizienten mit gespeicherten Koeffizienten
→ (Wieder-) Erkennung des ähnlichsten Gesichtes
- Auf diese Weise: Klassifikation mittels PCA

Gesichter-Stichprobe



Eigenfaces = die Eigenvektoren

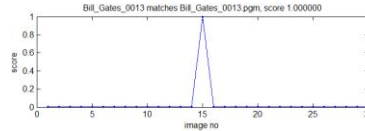


Principal Component Analysis: Eigengesichter

26

Gesichtersuche

- Suchgesicht in Stichprobe enthalten

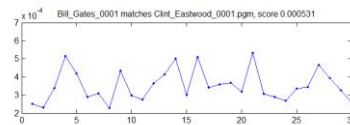


- nicht in Stichprobe enthalten

- Ähnlichkeit $\sim |\sum \alpha_i - \beta_i|$,
 α_j, β_j = Koeffizienten der verglichenen
Bilder in der Basis der (20 verwendete)
Eigenvektoren

- Hier als Ähnlichkeitsmaß benutzt:

$$\text{score} = \frac{1}{1 + |\sum \alpha_i - \beta_i|} \in [0, 1]$$



Übersicht

27

Merkmale

Principal Component Analysis:

Konzept

Formulierung

Anwendung

Eigengesichter

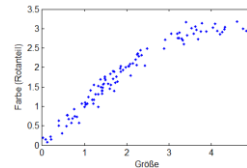
Ausblick

Hochdimensionale Daten

- Viele Merkmale = hochdimensionaler Merkmalsraum
- Numerische Eigenwertsuche einer $D \times D$ -Matrix hat Komplexität $\mathcal{O}(D^3)$, Rechenaufwand kann schnell zu groß werden
→ spezielle Algorithmen, die nicht alle, sondern nur die größten Eigenwerte suchen

Nichtlineare Komponenteanalyse

- PCA ist nur eine lineare Transformation, damit nicht erfassbar sind nichtlineare Abhängigkeiten zwischen einzelnen Merkmalen (z.B. latente Variable = Reifegrad, Farbe geht linear mit Reifegrad, aber Größe sättigt)
- Projektion auf nichtlineare Mannigfaltigkeit ist kompliziert, aber möglich, z.B. mit
 - Neuronale Netze
 - Kernelverfahren

**Literatur**

- C.M.Bishop, "Pattern recognition and machine learning", Kapitel "Continuous latent variables"
- Implementierung des Eigenface-Algorithmus in Matlab:
<http://blog.cordiner.net/2010/12/02/eigenfaces-face-recognition-matlab/>

Eigenwerte und Eigenvektoren

- Definition: \vec{u} ist Eigenvektor der Matrix \vec{A} zum Eigenwert λ , wenn gilt
- $\vec{A}\vec{u} = \lambda\vec{u}$
- Berechnung: Eigenwerte sind Lösung der charakterist. Gleichung $\det(\vec{A} - \lambda\mathbf{1}) = 0$
- Anzahl Eigenwerte $\lambda_i \neq 0 = \text{Rang der Matrix } \vec{A} \leq \text{Dimension der Matrix}$
- Länge der Eigenvektoren $|\vec{u}|$ ist unbestimmt (sinnvoll: Normierung auf $|\vec{u}| = 1$)
- Zu einem ("entarteten") Eigenwert λ_i können auch mehrere Eigenvektoren gehören

- Eigenschaften:
 - $\text{Spur}(\vec{A}) = \sum \lambda_i$
 - $\det(\vec{A}) = |\vec{A}| = \prod \lambda_i$
- Besondere Eigenschaften für reelle, symmetrische Matrizen:
 - Eigenwerte sind reell
 - Eigenvektoren sind zueinander orthogonal: $\vec{u}_i^t \vec{u}_j = \delta_{ij} = \begin{cases} 1 & \text{für } i = j \\ 0 & \text{für } i \neq j \end{cases}$