Universität Potsdam
Institut für Informatik
Lehrstuhl Maschinelles Lernen

Maschinelles Lernen II
PCA

Christoph Sawade/Niels Landwehr
Tobias Scheffer

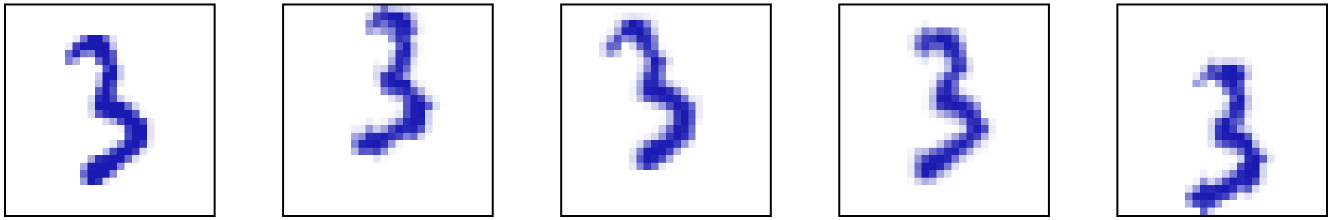
Überblick

- Principal Component Analysis
  - Optimierungsproblem
  - Adaption für hochdimensionale Daten
  - Kernel-PCA

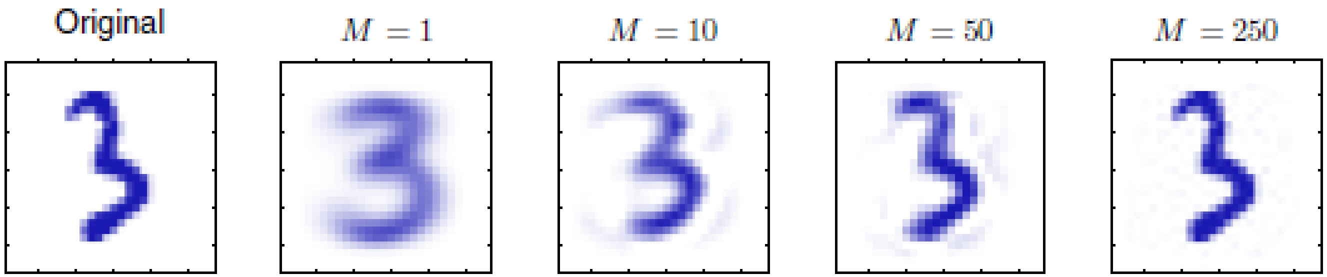
PCA
Motivation

- Datenkompression
- Preprocessing (Feature Selection / Noisy Feature)
- Datenvisualisierung

PCA
Beispiel

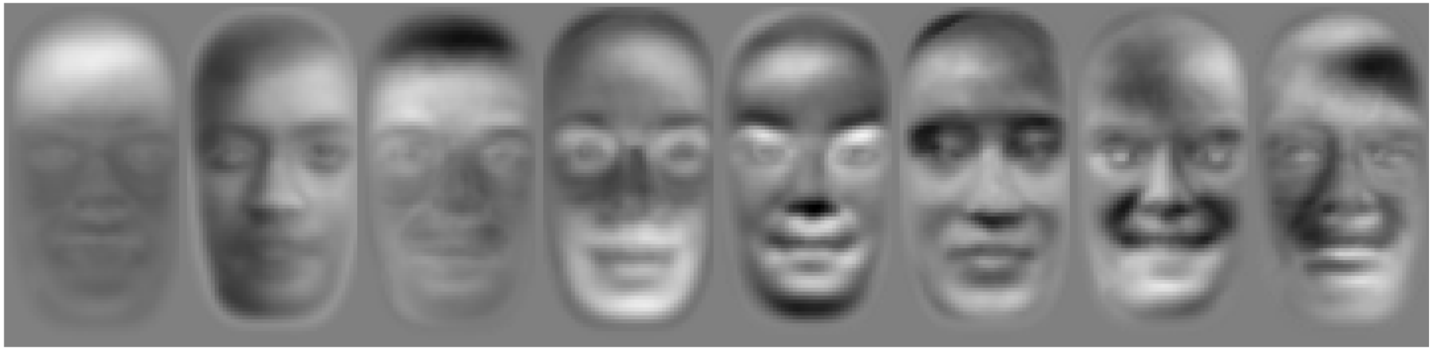


- Repräsentation von Digits als $m \times m$-Pixelmatrix
  - Viele Feature
    - sind aussagelos
    - ergeben sich aus anderen (weniger Freiheitsgrade)
- Ziel: Reduktion auf d-dimensionalen Hauptunterraum



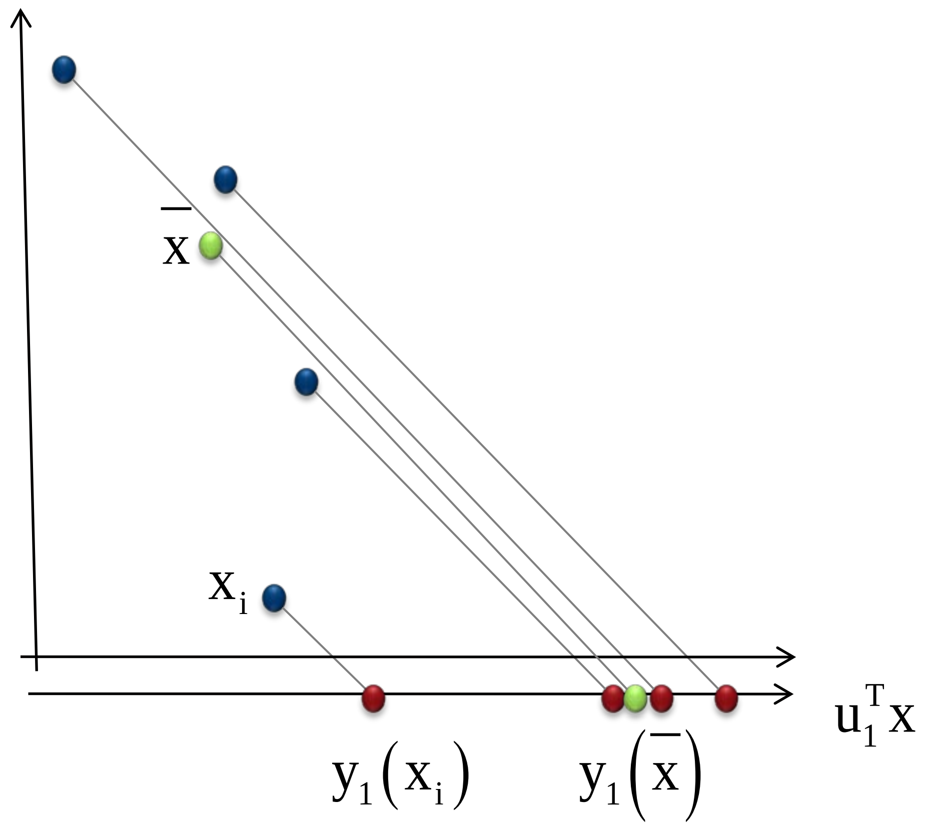
PCA
Beispiel

- Repräsentation von Gesichtern als $m \times m$-Pixelmatrix
  - Viele Feature
    - sind aussagelos
    - ergeben sich aus anderen (weniger Freiheitsgrade)
- Ziel: Reduktion auf d-dimensionalen Hauptunterraum



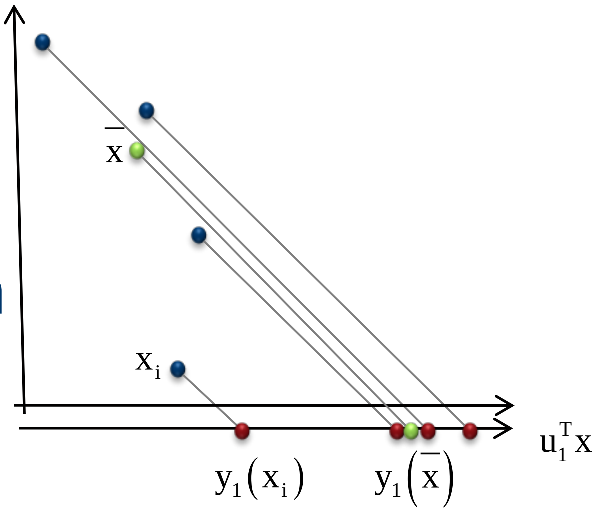
PCA
Projektion

- Eine Projektion ist eine idempotente lineare Abbildung



PCA
Projektion

- Eine Projektion ist eine idempotente lineare Abbildung



- Sei $u_1 \in \mathbb{R}^m$ mit $u_1^T u_1 = 1$
- $y_1(x) = u_1^T x$ stellt Projektion in einen eindimensionalen Unterraum dar

- Für Daten im Projektionsraum gilt:
  - Mittelpunkt: $y_1(\bar{x}) = u_1^T \bar{x}$
  - Varianz: $\frac{1}{n}\sum_{i=1}^{n}\left(u_1^T x_i - u_1^T \bar{x}\right)^2 = u_1^T \Sigma u_1$

PCA
Optimierungsproblem

- Ziel: Varianz der projizierten Daten $u_1^T \Sigma u_1$ soll nicht verloren gehen

- Maximiere $u_1^T \Sigma u_1$ bzgl. u_1, wobei $u_1^T u_1 = 1$
  - Lagrangian: $u_1^T \Sigma u_1 + \lambda_1 \left(1 - u_1^T u_1\right)$

PCA
Optimierungsproblem

- Ziel: Varianz der projizierten Daten $u_1^T \Sigma u_1$ soll nicht verloren gehen

- Maximiere $u_1^T \Sigma u_1$ bzgl. u_1, wobei $u_1^T u_1 = 1$
  - Lagrangian: $u_1^T \Sigma u_1 + \lambda_1 \left(1 - u_1^T u_1\right)$

- Ableiten, Nullsetzen:
  - $\Sigma u_1 = \lambda_1 u_1$ … Lösung muss Eigenvektor sein
  - $u_1^T \Sigma u_1 = \lambda_1$ … Varianz ist entsprechender Eigenwert
  - ➢ Bestimmung des größten Eigenwert

➢ Größter Eigenvektor ist erste Hauptkomponente

PCA

- Projektion von x in den Eigenraum

$$\mathrm{y}_1(\mathrm{x}) = \mathrm{u}_1^{\mathrm{T}}\mathrm{x} \Rightarrow \mathrm{y}(\mathrm{x}) = \mathrm{U}^{\mathrm{T}}\mathrm{x} \quad \text{mit} \quad \mathrm{U} = \begin{pmatrix} - & \mathrm{u}_1 & - \\ & \vdots & \\ - & \mathrm{u}_d & - \end{pmatrix}$$

- Größter Eigenvektor ist erste Hauptkomponente
- Folgende Hauptkomponenten sind orthogonale Richtungen die (Rest-) Varianz maximieren
- ➢ d Hauptkomponenten… Vektoren der d größten Eigenwerte

PCA
Rückprojektion

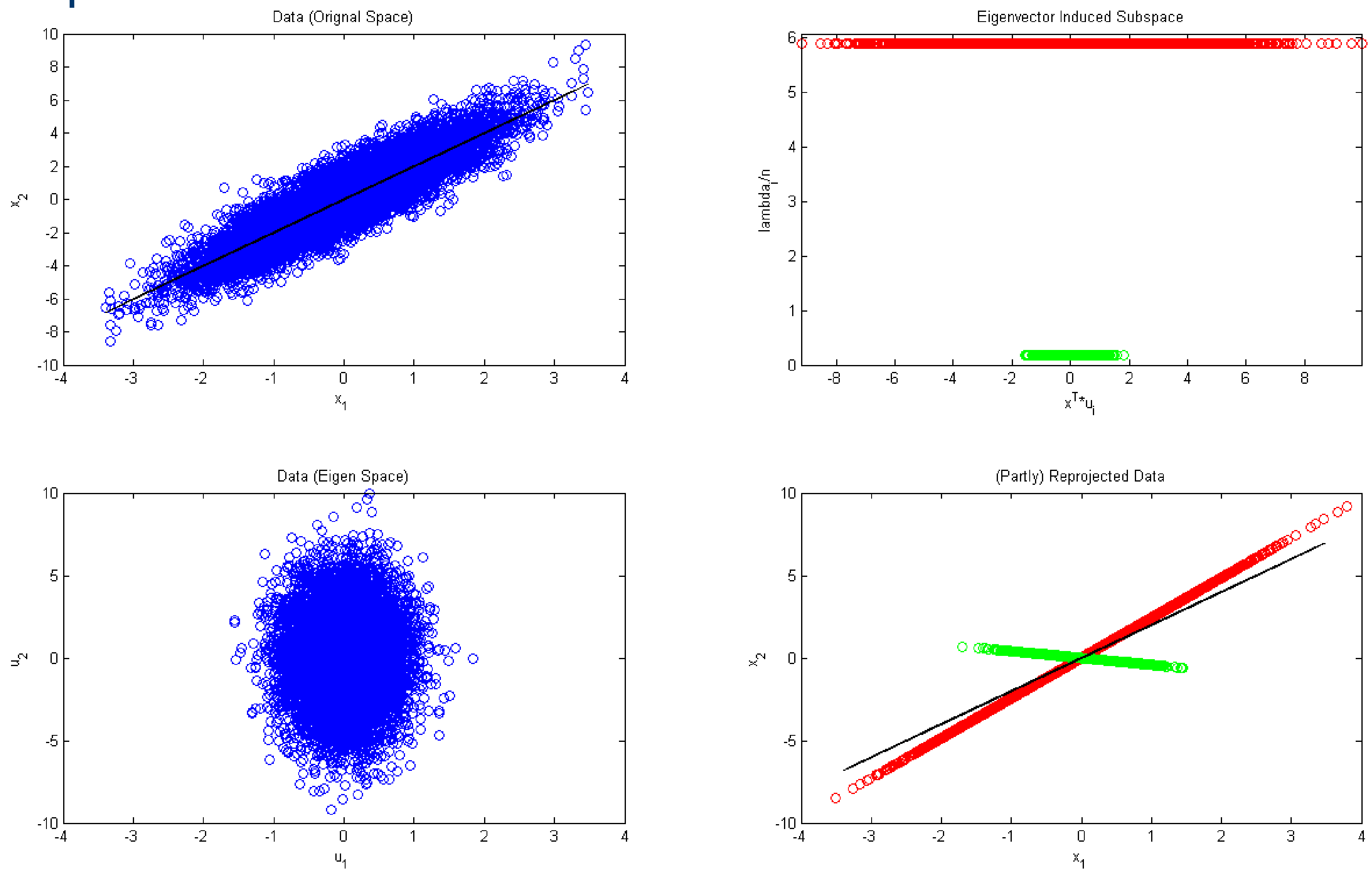
- Beobachtung:

$$x_i = \sum_{j=1}^{n} \left(x_i^T u_j \right) u_j \quad \left(X^T = X^T U U \right)$$

- Reduktion auf d Hauptkomponenten:

$$x_i = \sum_{j=1}^{d} \left(x_i^T u_j \right) u_j \quad \left(X^T = X^T V V, \ \forall k \geq d: v_{k\bullet} = 0^T \right)$$

PCA
Beispiel



PCA
hochdimensionale Daten

- Berechnung von d Eigenvektoren für m-dimensionale Daten ist $\mathcal{O}\left(\mathrm{dm}^2\right)$
  - Nicht berechenbar für große m
- Idee: Beispiele spannen einen linearen Unterraum mit höchsten $\mathrm{n}-1$ Dimensionen auf
- Sei $\overline{\mathrm{x}}=0$, dann gilt mit Hilfe der Datenmatrix $\mathrm{X} \in \mathbb{R}^{\mathrm{n} \times \mathrm{m}}$

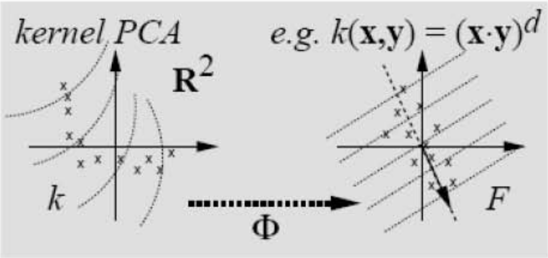
Eigenwertproblem

$$\Sigma \mathrm{u}_1 = \lambda_1 \mathrm{u}_1 \quad \Rightarrow \quad \left(\mathrm{n}^{-1}\mathrm{X}\mathrm{X}^{\mathrm{T}}\right)\mathrm{v}_1 = \lambda_1 \mathrm{v}_1, \quad \mathrm{v}_1 = \mathrm{X}\mathrm{u}_1$$

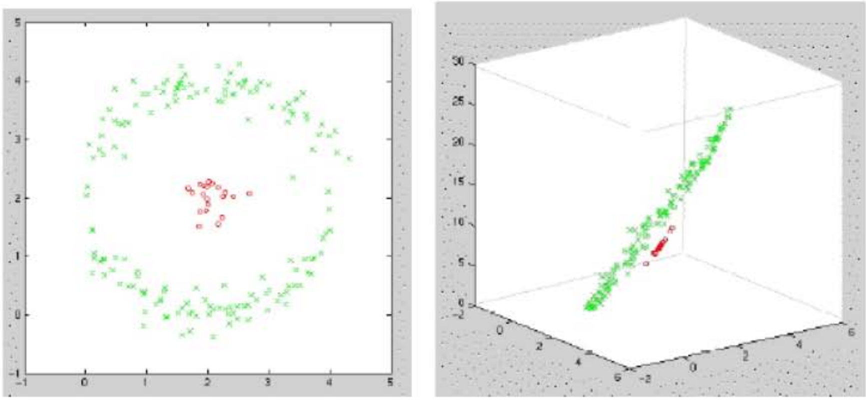
- Berechnung in $\mathcal{O}\left(\mathrm{dn}^2\right)$
- Lösung hat gleiche $\mathrm{n}-1$ Eigenwerte $\mathrm{u}_\mathrm{i} = \left(\mathrm{n}\lambda_\mathrm{i}\right)^{-1/2}\mathrm{X}^{\mathrm{T}}\mathrm{v}_\mathrm{i}$
  - bis auf Eigenwerte 0

Kernel-PCA

- Im Merkmalsraum nichtlineare Dimensionen.



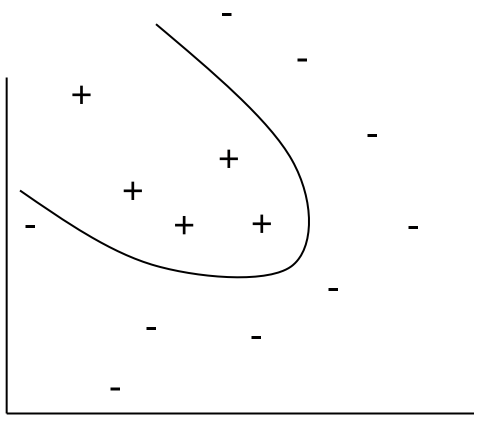
$$\Phi : \mathcal{X} = \mathbb{R}^2 \rightarrow \mathcal{H} = \mathbb{R}^3$$
$$(x_1, x_2) \mapsto (x_1, x_2, x_1^2 + x_2^2)$$

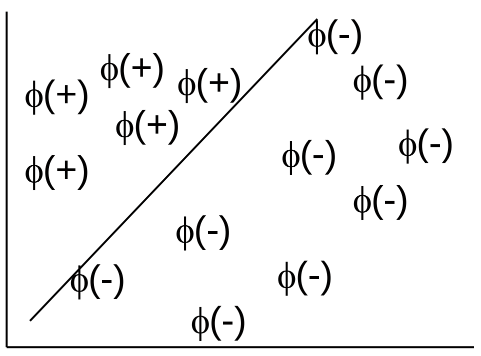


- Vorraussetzung: Daten gehen nur als inneres Produkt ein

Kernel-PCA
Wiederholung: Kerne

- Lineare Klassifikatoren:
  - Oft adäquat, aber nicht immer.
- Idee: Beispiele in anderen Raum abbilden, in dem sie linear klassifizierbar sind.
- Abbildung
  - $\mathbf{x} \mapsto \phi(\mathbf{x})$
- Zugehöriger Kernel
  - $k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^{\mathrm{T}} \phi(\mathbf{x}_j)$
- Kernel = Inneres Produkt = Ähnlichkeit der Beispiele.





Kernel-PCA

- Für $\sum_{i=1}^{n} \varphi(x_i) = 0$, lässt sich Eigenvektorproblem äquivalent umformen:

$$\Sigma u_i = \lambda_i u_i \quad \Rightarrow \quad K\alpha_i = \lambda_i n \alpha_i$$

- Projektion: $y_i(x) = \phi(x)^T v_i = \sum_{j=1}^{n} \alpha_{i,j} k(x, x_j)$

Zusammenfassung

- Ziel: Reduktion / Kompression von Daten auf wesentliche Komponenten
- Maximierung der Varianz führt zu Eigenwertproblem
- Hoch für hochdimensionale anwendbar
- Nicht-lineare Varianzkomponenten: Kernel-PCA
- Klassenabhängige Varianzminimierung führt zur Kernel Fisher Diskriminante