



Principal Component Analysis

Algorithmic Learning 64-360, part 3c

Norman Hendrich

University of Hamburg
MIN Faculty, Dept. of Informatics
Vogt-Kölln-Str. 30, D-22527 Hamburg
hendrich@informatik.uni-hamburg.de

22/06/2011



Gliederung

Dimensionsproblem

Wiederholung: Eigenvektoren

Principal Component Analysis

PCA und Neuronale Netze

Independent Component Analysis



Übersicht

- ▶ Mehrdimensionale Daten
- ▶ Das Problem der Dimensionalität
 “Curse of dimensionality”
- ▶ Wiederholung: Lineare Algebra
- ▶ Principal Component Analysis
- ▶ Neuronale Netze und PCA
- ▶ Independent Component Analysis



The *Curse of Dimensionality*

Begriff eingeführt von Richard Bellman, 1961

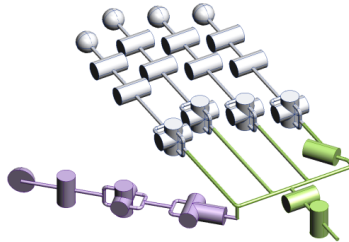
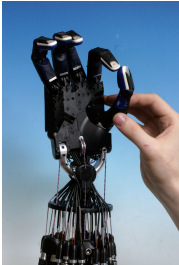
- ▶ Problem: Analyse/Klassifikation mehrdimensionaler Daten explodiert mit der Dimension n der Daten
- ▶ Volumen eines n -dim. Raums steigt exponentiell mit n
- ▶ Beispiele:
 - ▶ 1D: 100 Punkte reichen aus, um das Einheitsintervall $[0 \dots 1]$ mit einer Genauigkeit von 0.01 abzudecken
 - ▶ 10D: 10^{20} Punkte notwendig, um den 10D-Einheits-Hyperwürfel mit einer Genauigkeit von 0.01 abzudecken.
 - ▶ in anderen Worten: der 10-D Hyperwürfel ist um einen Faktor 10^{18} größer als das Einheitsintervall.



Curse of Dimensionality: Beispiele

- ▶ Roboterarm: typisch 6-7 DOF, also 6-7 Dimensionen
- ▶ menschliche Hand: Finger-Kinematik hat 24 Dimensionen, Griffe zusätzlich mit 6-DOF pro Kontakt zum Objekt
- ▶ VGA-Video: $640 \cdot 480 \cdot 3 = 921600$ Dimensionen
- ▶ QVGA, Graustufen: $320 \times 240 = 76800$ Dimensionen
- ▶ Audioclips: 16-bit, 44.1 kHz, 10 sec: 441000 Dimensionen
- ▶ 10-Mpixel Foto, RGB: $30 \cdot 10^6$ Dimensionen
- ▶ Gensequenz: typisch $10^6..10^9$ Basenpaare
- ▶ Textanalyse: Tausende Wörter...
- ▶ etc.

The Curse of Dimensionality



- ▶ Hand-Modell: Finger-Kinematik mit 24 Dimensionen
- ▶ Griffe zusätzlich mit 6-DOF pro Kontaktpunkt zum Objekt
- ▶ aber nicht alle Bewegungen unabhängig voneinander

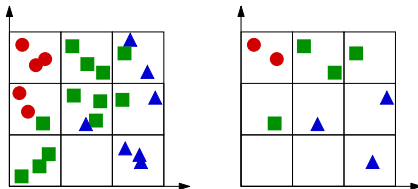


Beispiel: Sortierung in drei Klassen, 1D



- ▶ Eingabe-Raum in gleiche Intervalle einteilen
- ▶ Ergebnisse pro Klasse pro Intervall auszählen
- ▶ alle Testdaten in Intervalle einsortieren
- ▶ und die dominante Klasse pro Intervall berechnen
- ▶ aber: zu viele Überlappungen.

Beispiel: Sortierung in drei Klassen, 2D

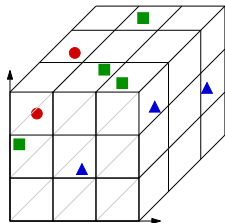


- ▶ Eingabe-Raum in gleiche Intervalle einteilen
- ▶ jetzt mit zwei unabhängigen Achsen

- ▶ bei gleicher Anzahl der Messwerte? (rechts)
- ▶ bei gleicher Dichte der Messwerte? (links)



Beispiel: Sortierung in drei Klassen, 3D



- ▶ Eingabe-Raum in gleiche Intervalle einteilen
- ▶ Übergang zu 3D oder nD macht das Problem deutlicher
- ▶ gleiche Anzahl der Messwerte: Eingaberaum fast leer
- ▶ gleiche Dichte der Messwerte: extrem viele Werte nötig



Volumen von Einheitswürfel und -kugel

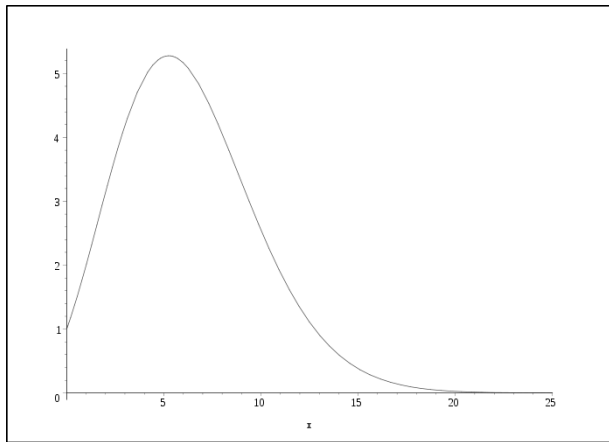
- ▶ Einheitswürfel: n -dimensionaler Würfel mit Seitenlänge 1
- ▶ $V_c(n) = 1^n$
- ▶ Einheitswürfel um die Einheitskugel: $V_{c2}(n) = 2^n$

- ▶ Einheitskugel: n -dimensionale Kugel mit Radius 1
- ▶ $V_s(n) = \frac{\pi^{n/2}}{\Gamma(1+n/2)}$

- ▶ Beispiel $n = 10$: $V_{c2}(10) = 1024$, $V_s(10) = \pi^5/120 \approx 2.55$
- ▶ mit wachsendem n wird die Kugel winzig relativ zum Würfel
- ▶ fast alle Punkte sind weit weg vom Mittelpunkt
- ▶ der n -dimensionale Raum besteht fast nur aus den Ecken/Rändern des Würfels



Relatives Volumen der Einheitskugel



(Wikipedia: Ball volume in n dimensions)



Lösungsvorschläge?

- ▶ Vorwissen über das Problem ausnutzen
- ▶ geeignete Vorverarbeitung der Messwerte
- ▶ Dimensionalität reduzieren

- ▶ Maximalanzahl der sinnvollen Features/Eigenschaften
- ▶ bei vorgegebener Anzahl der Messwerte

- ▶ darüber hinaus: Verlust an Genauigkeit durch Dimensionsproblem
- ▶ Auswertung/Lernen scheitert an geringer Dichte der Samples



Hoffnung: tatsächliche Komplexität gering

- ▶ Repräsentation eines Problems definiert die Dimensionalität
- ▶ tatsächliche Komplexität kann aber geringer sein

- ▶ Beispiel: Federpendel mit Mehr-Kamerasystem aufnehmen
- ▶ tatsächliche Bewegung ist 1D/2D/3D-Schwingung
- ▶ Bewegungsgleichungen bekannt und lösbar

- ▶ Sensordaten aus mehreren Kameras
- ▶ unterschiedliche Perspektiven, Pendel und Hintergrund
- ▶ Also: Extraktion der wesentlichen Parameter/Dimensionen aus der Vielzahl der Messwerte notwendig



Zusammenfassung

- ▶ Volumen des n -dimensionalen Raums wächst exponentiell
- ▶ exponentiell viele Messdaten, um n -dim. Raum auszufüllen

- ▶ prinzipiell exponentielles Wachstum der Komplexität der Zielfunktion mit n
- ▶ entsprechend auch exponentielle Anzahl der Testmuster zum Lernen der Funktion notwendig

- ▶ in 1D viele verschiedene Dichtefunktionen bekannt
- ▶ aber in nD normalerweise nur Gaussfunktionen analysierbar

- ▶ Hoffnung: tatsächliche Komplexität hängt nicht von allen Dimensionen ab: Dimensionsreduktion



Eigenwerte und Eigenvektoren

- ▶ Matrix **A**
- ▶ Eigenvektor x mit Eigenwert λ , wenn gilt:

$$\mathbf{A}x = \lambda x$$

- ▶ natürlich mit $x \neq 0$
- ▶ lineare Skalierung möglich

Eigenwerte und Eigenvektoren

- ▶ Umstellung des Problems ergibt die folgende Gleichung

$$(\mathbf{A} - \mathbf{I}\lambda)x = 0$$

$$\det|\mathbf{A} - \mathbf{I}\lambda| = 0$$

- ▶ Determinante ergibt Polynom n -ten Grades
- ▶ bis zu n reelle Nullstellen (Eigenwerte)
- ▶ immer n komplexe Nullstellen (Eigenwerte)



Singulärwertzerlegung

Singular Value Decomposition, SVD

- ▶ Darstellung einer Matrix als Produkt dreier spezieller Matrizen

$$M = U\Sigma V^*$$

- ▶ U eine unitäre $m \times m$ -Matrix
- ▶ V^* die Adjungierte einer unitären $n \times n$ -Matrix
- ▶ Σ eine durch M eindeutig bestimmte reellwertige $m \times n$ -Diagonalmatrix
- ▶ mit den Singulärwerten von M in der Diagonale
- ▶ es gilt $\sigma_1 \geq \dots \sigma_r > 0$.
- ▶ Lösung mit stabilen numerischen Verfahren



Principal Component Analysis

- ▶ einfache mathematische Transformation
- ▶ Drehung/Translation des Koordiantensystems
- ▶ um korrelierte Variablen durch unkorrelierte Variablen zu ersetzen
- ▶ neue Variablen nach Relevanz sortieren
- ▶ unwichtige Variablen weglassen: Reduktion der Dimensionalität

- ▶ Karhunen-Loève Transformation
- ▶ Hotelling-Transformation
- ▶ *proper orthogonal decomposition*

(Wikipedia: en.wikipedia.org/wiki/Principal_component_analysis, Karl Pearson 1901)



Grundidee

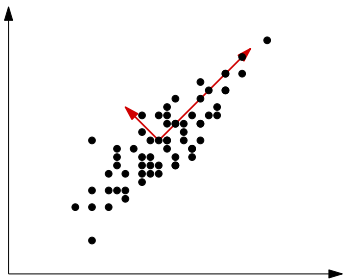
- ▶ Wahl eines neuen Koordinatensystems für die Daten
- ▶ Schritt 1: die Mittelwerte berechnen und abziehen
- ▶ Schritt 2: das Koordinatensystem drehen
- ▶ Schritt 3: Koordinaten nach der Varianz der Dimension sortieren

- ▶ Berechnung über Eigenwerte der Kovarianzmatrix
- ▶ bzw. Singulärwertzerlegung der Datenmatrix

- ▶ Hoffnung: große Varianz enthält wesentliche Merkmale der Daten
- ▶ Koordinaten mit kleiner Varianz unwichtig: weglassen



Beispiel: 2D-Datensatz



- ▶ Daten liegen schräg zu den ursprünglichen Koordinaten
- ▶ neue Koordinate x_1 zeigt in Richtung der Verteilung
- ▶ zweite Koordinate senkrecht dazu



Lösung des Dimensionsproblems mittels PCA

Prinzip:

- ▶ Projektion des eigentlichen Problems auf einen Subraum, in dem die Varianzen der Klassen hervorgehoben werden.

Dimensionsreduzierung:

- ▶ ergibt sich aus der Projektion
- ▶ Im Subraum werden (meist) weniger Merkmale benötigt



Aufgabenstellung: Klassifikation

Aufgabenstellung:

- ▶ Trainingsdaten: Geben sei eine Menge von Bilddaten \vec{x} mit $\vec{x} = [x_1, x_2, \dots, x_t]$
- ▶ Klassen: Die Bilder lassen sich in k - Klassen einteilen.
- ▶ Klassifikation: Es sollen ähnliche Bilder anhand der Trainingbilder klassifiziert werden.



Probleme der Klassifikation

Problem:

- ▶ Bilder sind meist größer als 76800 Pixel \Rightarrow direkter Vergleich zwischen Bildern aufwendig (Dimensionsproblem)
- ▶ Keine Generalisierung



PCA im Detail (1)

- ▶ Daten-Aquirierung:
Für den Aufbau des Subraumes müssen zunächst Trainingsdaten (\vec{x}_M) gesammelt werden.
- ▶ Mittelwertbereinigung:
Damit die PCA korrekt funktioniert müssen alle Trainingsdaten Mittelwertbereinigt sein.

$$\vec{x} - \vec{\mu} \quad \text{mit} \quad (1)$$

$$\vec{\mu} = \frac{1}{M} \sum_{i=1}^M \vec{x}_M \quad (2)$$



PCA im Detail (2)

- Berechnung der Kovarianz:

Aus den mittelwertbereinigten Daten wird die Kovarianzmatrix

$$\mathbf{Q} = \mathbf{P}\mathbf{P}^T \quad \text{mit} \quad (3)$$

$$\mathbf{P} = [\vec{x}_1 - \vec{\mu}, \vec{x}_2 - \vec{\mu}, \dots, \vec{x}_M - \vec{\mu}] \quad (4)$$

berechnet.

- Berechnung der Eigenwerte:

Die Eigenwerte der Kovarianzmatrix lassen sich durch folgende Gleichung bestimmen:

$$\lambda_i \vec{e}_i = \mathbf{Q} \vec{e}_i \quad \text{mit} \quad (5)$$

$$\lambda_i \quad i = 1 \dots m \quad \text{Eigenwerten und}$$

$$\vec{e}_i \quad i = 1 \dots m \quad \text{Eigenvektoren}$$



PCA im Detail (3)

- ▶ Dimensionsreduktion :
Sortiere die \vec{e}_i so das gilt :

$$\lambda_1 > \lambda_2 > \dots > \lambda_m$$

Somit nimmt der Informationsgehalt mit Index i ab.
 Für die Dimensionsreduktion werden nur die ersten n
 Eigenvektoren benutzt ($n < m$) da in ihnen die meiste
 Information enthalten ist.



PCA im Detail (4)

- ▶ Transformationsmatrix:
Mit der Transformationsmatrix

$$\mathbf{A} = (\vec{e}_1 \dots \vec{e}_n)^T$$

lassen sich die Eingangsdaten in den Eigenraum transformieren:

$$\vec{p}_i = \mathbf{A} \cdot \vec{x}_i \quad \dim(\vec{p}_i) = n$$

- ▶ Rückprojektion:
Da \mathbf{A} quadratisch und orthogonal gilt:

$$\mathbf{A}^{-1} = \mathbf{A}^T$$

Somit ergibt sich die Rücktransformation durch:

$$\vec{x}_i = \mathbf{A}^{-1} \vec{p}_i = \mathbf{A}^T \vec{p}_i$$



Dimensionsbestimmung

- ▶ Wird n groß gewählt
 ⇒ wenig Informationsverlust aber auch keine große Dimensionsreduktion
- ▶ Wird n klein gewählt
 ⇒ u.U. viel Informationsverlust aber nachfolgende Operationen einfacher

Automatische Wahl von n :

$$\frac{\sum_{i=1}^n \lambda_i}{\sum_{i=1}^m \lambda_i} \geq T$$

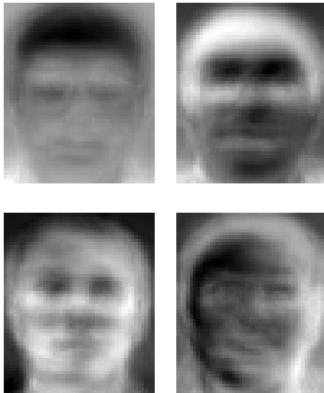


Beispiel: Gesichtserkennung mit Eigenfaces

- ▶ Graustufenbilder aller zu erkennenden Personen
- ▶ Vorverarbeitung zum Angleichen der Bilder
- ▶ Anwendung der PCA zur Identifikation der relevanten Merkmale
- ▶ Eigenfaces: die “mittleren” Gesichter zu den Eigenwerten
- ▶ Erkennung durch Transformation des Eingabebildes
- ▶ Vergleich mit den transformierten aller bekannten Referenzbilder

(M. Turk & A. Pentland, Eigenfaces for recognition, Journal of Cognitive Neuroscience 3, 71–86, 1991)

Eigenfaces





Implizite Kovarianz (1)

Berechnung der Kovarianzmatrix problematisch:

Bild von 320×240 Pixeln \Rightarrow Vektor mit der Dimension 76800

Kovarianzmatrix :

$$\mathbf{Q} = \mathbf{P}\mathbf{P}^T \quad \mathbf{Q} \in M_{t \times t}$$

$\Rightarrow 76800^2 = 5,89824 * 10^9$ Elemente

\Rightarrow bei einem Byte pro Element $\approx 5,5$ GB

\Rightarrow entsprechend schlimmer bei noch höherem n



Implizite Kovarianz (2)

Normalerweise hat man aber nur $M \ll \dim(\vec{x}_i)$ Eingangsdaten und somit kann man auch höchstens M Eigenvektoren finden.

Implizite Kovarianz:

$$\tilde{\mathbf{Q}} = \mathbf{P}^T \mathbf{P} \quad \tilde{\mathbf{Q}} \in M_{M \times M}$$

Bei Beispielsweise 100 Bildern als Input:

$\Rightarrow 100^2 = 10000$ Elemente

\Rightarrow bei einem Byte pro Element $\approx 9,77\text{kb}$



Eigenwerte und Eigenvektoren der impliziten Kovarianzmatrix

Die Eigenwerte von \mathbf{Q} und ihre korrespondierenden Eigenvektoren lassen sich aus den Eigenwerten und -vektoren von $\tilde{\mathbf{Q}}$ berechnen:

$$\begin{aligned}\lambda_i &= \tilde{\lambda}_i \\ \vec{e}_i &= \tilde{\lambda}_i^{-\frac{1}{2}} \mathbf{P} \tilde{\vec{e}}_i\end{aligned}$$



Klassifikation und PCA (1)

Beispiel aus „Turk and Pentland: Eigenfaces for Recognition “

- ▶ Projektion der Klassen in den Eigenraum:

$$\vec{\Omega}_c = \mathbf{A}^T (\vec{x}_c - \vec{\mu}) \quad c = 1, \dots, k \quad (6)$$

- ▶ Bestimme maximalen Abstand zwischen Klassen:

$$\theta_l = \frac{1}{2} \max_{j,k} \{ \|\vec{\Omega}_j - \vec{\Omega}_i\| \} \quad j, i = 1, \dots, k \quad (7)$$

Klassifikation und PCA (2)

- ▶ Klassifikation eines neuen Bildes \vec{x} :
 - ▶ Projektion in den Eigenraum :

$$\vec{\Omega} = \mathbf{A}^T (\vec{x} - \vec{\mu}) \quad (8)$$

- ▶ Klassenabstand bestimmen:

$$\epsilon_c = \|\vec{\Omega} - \vec{\Omega}_c\| \quad (9)$$

- ▶ Bestimmung des Abstands zwischen Eingabe und Rückprojektion:

$$\epsilon = \|\vec{x} - \vec{x}_r\| \quad \text{mit} \quad (10)$$

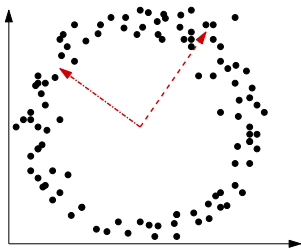
$$\vec{x}_r = \mathbf{A}\vec{\Omega} + \vec{\mu} \quad (11)$$

Klassifikation und PCA (3)

► Klassifikation

- Falls $\epsilon \geq \theta_I$
 Input ist kein Gesicht.
- Falls $\epsilon < \theta_I$ und $\forall c, \epsilon_c \geq \theta_I$
 Input ist ein unbekanntes Gesicht.
- Falls $\epsilon < \theta_I$ und $\epsilon_{c^*} = \min_c \{\epsilon_c\} < \theta_I$
 Input enthält ein Gesicht von Person c^*

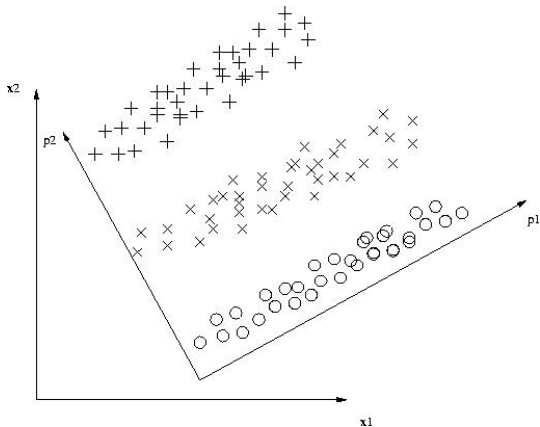
Problemfälle für PCA



- ▶ Daten werden nach Varianz sortiert
- ▶ implizite Annahme einer Gauss-Verteilung
- ▶ aber ungeeignet für Daten mit anderer Verteilung
- ▶ z.B. bi-modale oder multi-modale Verteilungen

Problemfälle für PCA

Adidas-Problem:





Kernel PCA

- ▶ Kombination von nicht-linearer Transformation mit anschließender linearer PCA
- ▶ Einsatz von Kernel-Funktionen
 - ▶ Polynomkernel
 - ▶ Gauss-Kernel
 - ▶ sigmoid, usw.
- ▶ gleiche Idee wie bei der Support-Vektor Maschine

PCA und Neuronale Netze

Die erste Hauptkomponente der PCA lässt sich auch über die Hebb-Regel von einem einschichtigen Perzeptronnetzwerk lernen.
 Regel von Yuille et al. :

$$\vec{y} = \sum \omega_i \xi_j = \mathbf{w}^T \vec{x} = \vec{x}^T \mathbf{w}$$

$$\Delta \omega_j = \eta (V x_j - \omega_j |\mathbf{w}|^2)$$

Der Gewichtsvektor \vec{w} zeigt im Konvergenzfall in die Richtung des Eigenvektors der Kovarianzmatrix mit der größten Varianz.



Oja-Algorithmus

- ▶ *start*: Menge X von n -dimensionalen Eingabevektoren
- ▶ Vektor w zufällig initialisiert ($w \neq 0$)
- ▶ Lernrate γ mit $0 < \gamma \leq 1$

- ▶ *update*: wähle zufälligen Vektor x aus X
- ▶ berechne Skalarprodukt $\Phi = x \cdot w$
- ▶ neuer Gewichtsvektor ist $w + \gamma\Phi(x - \Phi w)$
- ▶ gehe zu *update*, reduziere γ

(Oja 1982, Rojas 5.3.1)



Independent Component Analysis

- ▶ ICA Webseite: <http://www.cis.hut.fi/projects/ica/>

Independent Component Analysis (1)

Folgender Sachverhalt verdeutlicht die Arbeitsweise der Independent Component Analysis (kurz ICA):

In einem Raum sind zwei Lautsprecher aufgestellt, die zwei verschiedene Tonsignale $s_1(t)$ und $s_2(t)$ ausgeben.

Die zwei Tonsignale werden von zwei verschiedenen Mikrofonen an verschiedenen Stellen aufgenommen.



Independent Component Analysis (2)

Die beiden Mikrofone nehmen zwei unterschiedliche Mischungen $x_1(t)$ und $x_2(t)$ der Originaldaten auf, wobei:

$$x_1(t) = a_{11}s_1(t) + a_{12}s_2(t) \quad (12)$$

$$x_2(t) = a_{21}s_1(t) + a_{22}s_2(t) \quad (13)$$

Die Faktoren a_{11} , a_{12} , a_{21} , a_{22} und die Originaldaten sind aus der Perspektive der Mikrofone unbekannt. In Vektornotation kann die obige Gleichung allgemein wie folgt geschrieben werden:

$$\vec{x}(t) = \mathbf{A} \cdot \vec{s}(t) \quad (14)$$

wobei $\vec{x}(t), \vec{s}(t) \in R^n$ sind und \mathbf{A} eine $n \times n$ -Matrix ist.



Independent Component Analysis (3)

Falls die Originaldaten folgenden Bedingungen genügen, kann die Mischungsmatrix \mathbf{A} mit der ICA bestimmt werden:

1. Die Originaldaten müssen statistisch unabhängig sein
2. Die Originaldaten müssen stationär sein
3. Maximal eine Originalquelle darf gaußverteilt sein

Die Originaldaten können aus der inversen Mischungsmatrix $\mathbf{W} = \mathbf{A}^{-1}$ der Matrix \mathbf{A} berechnet werden:

$$\mathbf{s}(t) = \mathbf{W} \cdot \mathbf{x}(t) \quad (15)$$



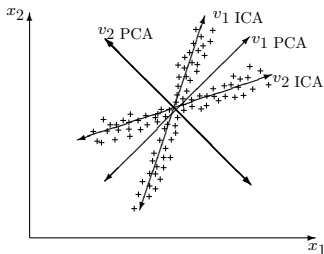
Independent Component Analysis (4)

Die Wiederherstellung der ursprünglichen Daten unterliegt allerdings zwei Einschränkungen:

1. Die Energie oder die Varianzen der einzelnen Originalquellen können nicht wiederhergestellt werden.
2. Die Reihenfolge der Originalsignale $s_i(t)$ kann ebenfalls nicht rekonstruiert werden.

Vergleich ICA mit PCA

Gezeichnet sind die Vektoren zweier korrelierter, nicht-normalverteilter Zeitserien. Die PCA projiziert auf eine Basis, deren Achsen orthogonal sind, wobei die 1. Achse in Richtung der größten Varianz zeigt. Die Achsen der ICA müssen nicht orthonormal sein, so daß die Varianz für beide Achsen maximiert werden kann. Dieses führt zu einer günstigeren Dekorrelation.



Dimensionsreduktion mit der ICA (1)

Bei der PCA ist die Reihenfolge der zu selektierenden Principal Components durch die Größe der Eigenwerte vorgegeben. Für die gefundenen Independent Components existiert diese Reihenfolge nicht.

Es gibt deshalb verschiedene Ansätze, die ICA zur Dimensionsreduktion zu benutzen:

- ▶ Ordnung der Zeilen in der Mischungsmatrix A nach der euklidischen L_2 -Norm. Die Zeilen von A mit der größten L_2 -Norm haben die größte Energie und somit haben die Quellen die zu diesen Zeilen gehören, einen größeren Einfluß auf die beobachteten gemischten Signale $\mathbf{x}(t)$.



Dimensionsreduktion mit der ICA (2)

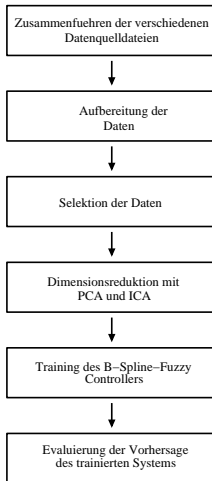
- ▶ Selektion der m Quellen mit der größten Amplitude, also der Komponenten $s_i(t)$ der Vektoren $\mathbf{s}(t)$ mit der größten L_∞ -Norm.
- ▶ Eine weitere Möglichkeit ist die Berechnung der Independent Component Analysis auf Principal Components. Bei diesem Verfahren werden die n -dimensionalen Eingangsvektoren $\mathbf{x}(t)$ mit Hilfe der PCA auf m -dimensionale Vektoren reduziert. Auf diesen reduzierten Vektoren wird dann die ICA berechnet. (Geschwindigkeitsvorteil).
- ▶ Als weitere Möglichkeit bietet sich Input Selection an, um die Komponenten mit den interessantesten Informationen zu selektieren.



Anwendungsgebiete ICA

- ▶ Filtern von MEG/EEG Daten
- ▶ Reduzierung von Rauschen in natürlichen Bildern
- ▶ Telekommunikation
- ▶ Auffinden von versteckten Faktoren in Finanzdaten

Aufbau des Vorhersagesystems





Vorhersage ohne fundamentale Daten (1)

Bei der ersten Testreihe soll für zwei verschiedene Paare von Trainings- und Testmengen das unterschiedliche Verhalten des Vorhersagesystems für die unterschiedlichen Methoden der Dimensionsreduktion analysiert werden.

Der Vorhersagehorizont h ist 5 und die drei Paare von Trainings- und Testmengen sind Standardintervalle. Für jeden Tag wurde der Vektor $\mathbf{x}(t)$ aus den Returns in Prozent $r_h(t)$ für den S&P 500 Index wie in Gleichung 16 berechnet.

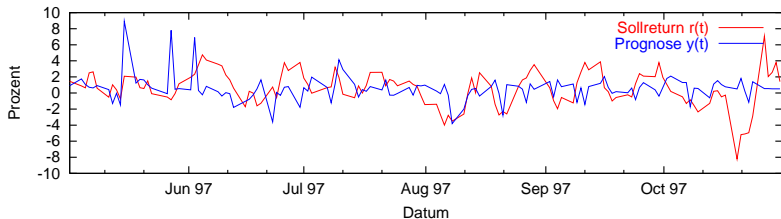
$$\mathbf{x}(t) = (r_{-1}(t), r_{-2}(t), \dots, r_{-50}(t))^T \quad (16)$$



Vorhersage ohne fundamentale Daten (2)

Als Dimensionsreduktionsverfahren werden in dieser Testreihe die PCA, die ICA mit der L_2 -Norm als Selektionskriterium gegenübergestellt.

Die Sollreturns $r(t)$ und die prognostizierten Returns $y(t)$ für die Vorhersage mit der ICA:





Vorhersage mit fundamentalen Daten (1)

Mit dieser Methode soll herausgefunden werden, ob sich die Vorhersagen, basierend auf der Zeitreihenanalyse mit ICA, durch Hinzunahme der fundamentalen Daten verbessern lassen.

Der Vektor $\mathbf{p}(t)$ besteht hier aus 6 mittels L_2 -Norm berechneten Independent Components, die aus k-Day>Returns des S&P 500 Index berechnet wurden.

Als 7. Komponente kommt zusätzlich der jeweils aktuelle Wert aus einer der 10 Zeitserien f_i mit den fundamentalen Daten hinzu. Der Vorhersagehorizont h ist 5 und der Vektor $\mathbf{x}(t)$ wie in Gleichung.

$$\mathbf{x}(t) = (r_{-1}(t), r_{-2}(t), \dots, r_{-50}(t), f_i(t))^T. \quad (17)$$



Vorhersage mit fundamentalen Daten (2)

Der Vektor $\mathbf{p}(t)$ wird mittels ICA aus den Return-Komponenten $r_{-i}(t)$ des Vektors $\mathbf{x}(t)$ berechnet. Die letzte Komponente $f_i(t)$ geht direkt in den Vektor $\mathbf{p}(t)$ ein.

In einer der folgenden Testreihen wird dann die Performance des Vorhersagesystems mit den Independent Components der Zeitreihenanalyse und den Kombinationen aller vielversprechenden Kandidaten untersucht.

Für das Vorhersagesystem wurden US-amerikanische Indikatoren und Aktienindizes verwendet. Das System sollte sich aber auch auf andere Märkte übertragen lassen, sofern für die jeweiligen Märkte genügend Datenmaterial öffentlich zugänglich ist.

Vorhersage mit fundamentalen Daten (3)

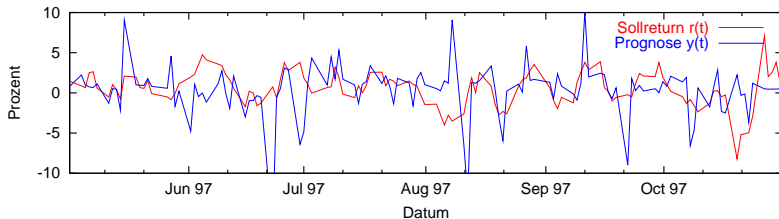
Das entwickelte Prognosemodell ermöglicht die Kombination von Zeitreihenanalyse mit der Analyse fundamentaler Daten. Die folgende Tabelle gibt eine Übersicht über die benutzten fundamentalen Daten:

Name	Kürzel
U.S. Weekly Leading Index	wli
U.S. Coincident Leading Index	usci
M3 Money Stocks	m3
Exchange Rate Swiss Franc/US\$	exszus
Consumer Price Index	cpiaucsl
Civilian Unemployment Rate	unrate
Consumer Sentiment Index	umcsent
Real Disposable Personal Income	dspic96
NAPM Manufact. Composite Index	napm
Manufacturers' New Orders	neworder

Überblick über die fundamentalen Daten

Vorhersage mit fundamentalen Daten (4)

Die Geldmenge M3 als zusätzlicher Eingang hat die Vorhersage auf allen drei Intervallen verbessert. Die Indikatoren “Manufacturers New Orders” (neworder), “Arbeitslosenquote” (unrate) und “Real Disposal Personal Income” (dspic96) verbessern die Vorhersage bezüglich des “Mean Profit per Trade” Kriteriums auf jeweils zwei der drei Intervalle.



Die Sollreturns $r(t)$ und die prognostizierten Returns $y(t)$ für die Vorhersage aus 6 Independent Components und den Auftragseingängen (neworder) als zusätzlichem Eingang auf der Testmenge des Intervalls 2.