

# **Analyse dünnbesetzter Hauptachsen für Frequenzdaten**

Tobias Bork

Geboren am 21. November 1997 in Reutlingen

15. Februar 2020

Bachelorarbeit Mathematik

Betreuer: Prof. Dr. Jochen Garcke

Zweitgutachter: Prof. Dr. X Y

MATHEMATISCHES INSTITUT FÜR NUMERISCHE SIMULATION

MATHEMATISCH-NATURWISSENSCHAFTLICHE FAKULTÄT DER  
RHEINISCHEN FRIEDRICH-WILHELMS-UNIVERSITÄT BONN



## *Danksagung*

The acknowledgments and the people to thank go here, don't forget to include your project advisor...



# Inhaltsverzeichnis

<b>Danksagung</b>	<b>1</b>
<b>1 Einführung</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Dimensionsreduktionsverfahren . . . . .	1
1.3 Sparse Approximations / Representations . . . . .	2
1.4 Interpretierbarkeit . . . . .	2
1.5 Compressed Sensing Beispiel . . . . .	2
<b>2 Mathematische Grundlagen</b>	<b>3</b>
2.1 Lineare Algebra . . . . .	3
2.1.1 Orthogonalität . . . . .	3
2.1.2 Matrixzerlegungen . . . . .	5
2.1.3 Matrix Approximation . . . . .	6
2.2 Analysis . . . . .	7
2.2.1 Norm . . . . .	7
2.3 Generalisierte lineare Modelle . . . . .	8
2.3.1 Grundlagen aus der Statistik . . . . .	8
2.3.2 Lineare Regression . . . . .	9
2.3.3 Ridge Regression . . . . .	11
2.3.4 LASSO . . . . .	12
2.3.5 Elastic Net . . . . .	14
2.3.6 Vergleich der Regressionsmethoden . . . . .	15
<b>3 Hauptkomponentenanalyse</b>	<b>19</b>
3.1 Konstruktion . . . . .	20
3.1.1 Problemformulierung als Varianzmaximierung . . . . .	20
3.1.2 Formulierung als Regressionsproblem . . . . .	22
3.1.3 Weitere Formulierungen . . . . .	24
3.2 Selektion der Hauptkomponenten . . . . .	25
3.3 Grenzen der Anwendbarkeit . . . . .	27
3.4 Erweiterungen der Hauptkomponentenanalyse . . . . .	28
3.5 Theoretische Aussagen . . . . .	28
<b>4 Dünnbesetzte Hauptkomponentenanalyse</b>	<b>31</b>
4.1 Problemformulierung . . . . .	31
4.2 Relaxation . . . . .	32
4.3 Konstruktion . . . . .	34
4.4 Anpassung der Transformation, Residuen und Varianzen . . . . .	35
4.5 Wahl der Hyperparameter . . . . .	37
4.6 Theoretische Aussagen . . . . .	39
<b>5 Implementierung</b>	<b>41</b>

5.1	Numerische Lösung	41
5.2	Algorithmus	42
5.3	Komplexität	43
5.4	Numerische Lösung im Fall $p \gg n$	44
5.5	Eigene Implementierung in Python	45
<b>6</b>	<b>Anwendung</b>	<b>47</b>
6.1	Beschreibung des Datensatzes	47
6.2	Vorverarbeitung der Daten	47
6.3	Anwendung auf Frequenzdaten	48
6.4	Auswertung der Ergebnisse	48
6.5	Vergleich zur klassischen Hauptkomponentenanalyse	49
6.6	Effekte der Hyperparameter	49
6.6.1	Zeit	49
6.6.2	Anzahl an Iterationen	49
6.6.3	Effekt auf Resultate	49
<b>7</b>	<b>Ausblick / Zusammenfassung</b>	<b>51</b>
7.1	Einsetzbarkeit	51
7.2	Übertragbarkeit	51
7.3	Ongoing Research / Weitere Techniken	51
	<b>Literatur</b>	<b>53</b>

## Kapitel 1

# Einführung

We are drowning in information and starving for knowledge. - John Naisbitt

Machine Learning, unsupervised, supervised Methoden

Zahlen und Fakten zu Big Data

Ab wann sprechen wir eigentlich von Big Data?

It should be noted, however, that even when one has an apparently massive data set, the effective number of data points for certain cases of interest might be quite small. In fact, data across a variety of domains exhibits a property known as the long tail, which means that a few things are very common, but most things are quite rare."(Murphy, Machine Learning)

## 1.1 Motivation

So ist man meist besonders an der Bildung sog. Cluster, also Gruppierungen, interessiert. Datenpunkte, die im entstehendem Bild nach Anwendung der Hauptkomponentenanalyse nah beieinander liegen, sind in gewisser Weise ähnlich zueinander während Datenpunkte, die weit von einander entfernt liegen, wenig Ähnlichkeit aufweisen. Abbildung CITE zeigt die Entstehung solcher Cluster auf dem Datensatz. Mit diesem Verfahren lässt sich daher eine Art Struktur in den Daten erkennen, die für weitere Analysezwecke ausgenutzt werden kann.

The goals of PCA are to

(1)

extract the most important information from the data table; (2)

compress the size of the data set by keeping only this important information; (3)

simplify the description of the data set; and (4)

analyze the structure of the observations and the variables.

## 1.2 Dimensionsreduktionsverfahren

CURSE OF DIMENSIONALITY (Bellman 1961)

High dimensionality means that the dataset has a large number of features. The primary problem associated with high-dimensionality in the machine learning field is model overfitting, which reduces the ability to generalize beyond the examples in the training set. Richard Bellman described this phenomenon in 1961 as the Curse of Dimensionality where "Many

algorithms that work fine in low dimensions become intractable when the input is high-dimensional. “

Let’s say that you want to predict what the gross domestic product (GDP) of the United States will be for 2017. You have lots of information available: the U.S. GDP for the first quarter of 2017, the U.S. GDP for the entirety of 2016, 2015, and so on. You have any publicly-available economic indicator, like the unemployment rate, inflation rate, and so on. You have U.S. Census data from 2010 estimating how many Americans work in each industry and American Community Survey data updating those estimates in between each census. You know how many members of the House and Senate belong to each political party. You could gather stock price data, the number of IPOs occurring in a year, and how many CEOs seem to be mounting a bid for public office. Despite being an overwhelming number of variables to consider, this just scratches the surface. TL;DR — you have a lot of variables to consider. If you’ve worked with a lot of variables before, you know this can present problems. Do you understand the relationships between each variable? Do you have so many variables that you are in danger of overfitting your model to your data or that you might be violating assumptions of whichever modeling tactic you’re using? You might ask the question, “How do I take all of the variables I’ve collected and focus on only a few of them?” In technical terms, you want to “reduce the dimension of your feature space.” By reducing the dimension of your feature space, you have fewer relationships between variables to consider and you are less likely to overfit your model. (Note: This doesn’t immediately mean that overfitting, etc. are no longer concerns — but we’re moving in the right direction!) Somewhat unsurprisingly, reducing the dimension of the feature space is called “dimensionality reduction.” There are many ways to achieve dimensionality reduction, but most of these techniques fall into one of two classes: Feature Elimination Feature Extraction

2. Why is Dimensionality Reduction required? Here are some of the benefits of applying dimensionality reduction to a dataset:

Space required to store the data is reduced as the number of dimensions comes down Less dimensions lead to less computation/training time Some algorithms do not perform well when we have a large dimensions. So reducing these dimensions needs to happen for the algorithm to be useful It takes care of multicollinearity by removing redundant features. For example, you have two variables – ‘time spent on treadmill in minutes’ and ‘calories burnt’. These variables are highly correlated as the more time you spend running on a treadmill, the more calories you will burn. Hence, there is no point in storing both as just one of them does what you require It helps in visualizing data. As discussed earlier, it is very difficult to visualize data in higher dimensions so reducing our space to 2D or 3D may allow us to plot and observe patterns more clearly

### 1.3 Sparse Approximations / Representations

### 1.4 Interpretierbarkeit

In Zeiten der Datenvielfalt erfährt die Interpretation dieser enorme Wichtigkeit.

### 1.5 Compressed Sensing Beispiel



## Kapitel 2

# Mathematische Grundlagen

In diesem Kapitel werden wir die wichtigsten mathematischen Grundlagen, die wir für die Formulierung der dünnbesetzten Hauptkomponentenanalyse benötigen, einführen. Dazu beschäftigen wir uns zunächst mit Grundbegriffen aus der linearen Algebra, Matrixzerlegungen und ausgewählten Approximationsproblemen in Abschnitt 2.1. Ein Großteil dieses Kapitels ist linearen Regressionsmodellen gewidmet, welche wir mit verschiedenen Straftermen versehen werden, um gewisse Effekte zu erzielen. Anhand eines Beispiel-Datensatzes werden wir in der Lage sein, diese Effekte visuell nachzuvollziehen. Für den weiteren Verlauf dieser Arbeit ist das Verständnis der Strafterme von entscheidender Bedeutung. Zu Schluss werden wir in Abschnitt ?? Grundlagen der Signalverarbeitung ausarbeiten, da wir in Kapitel 6 die dünnbesetzte Hauptkomponentenanalyse auf Frequenzdaten anwenden.

## 2.1 Lineare Algebra

Ein Großteil der Mathematik der Hauptkomponentenanalyse beruht auf Methoden der linearen Algebra. Aufgrund des Anwendungsfalls werden wir uns auf die Einführung der Grundbegriffe in reellen Vektorräumen beschränken.

### 2.1.1 Orthogonalität

**Definition 2.1** (Skalarprodukt [Jän07]). Sei  $V$  ein reeller Vektorraum. Ein *Skalarprodukt* in  $V$  ist eine Abbildung  $\langle \cdot, \cdot \rangle : V \times V \rightarrow \mathbb{R}$  mit den folgenden Eigenschaften:

- (i) Für jedes  $x \in V$  sind die Abbildungen

$$\begin{array}{ll} \langle \cdot, x \rangle : V \rightarrow \mathbb{R} & \langle x, \cdot \rangle : V \rightarrow \mathbb{R} \\ v \mapsto \langle v, x \rangle & v \mapsto \langle x, v \rangle \end{array}$$

linear. (Bilinearität)

- (ii)  $\langle x, y \rangle = \langle y, x \rangle$  für alle  $x, y \in V$  (Symmetrie)  
 (iii)  $\langle x, x \rangle > 0$  für alle  $x \neq 0$  (Positive Definitheit)

Allgemein versteht man unter einem *euklidischem Vektorraum* ein Paar  $(V, \langle \cdot, \cdot \rangle)$ , welches aus einem reellem Vektorraum  $V$  und einem Skalarprodukt  $\langle \cdot, \cdot \rangle$  auf  $V$  besteht. Durch das Skalarprodukt wird eine Norm auf  $V$  induziert:

$$\|v\| := \sqrt{\langle v, v \rangle}$$

In den folgenden Kapiteln werden wir uns vor allem mit dem *Standardskalarprodukt* im  $\mathbb{R}^n$  beschäftigen. Dies ist gegeben durch

$$\langle x, y \rangle = x_1 y_1 + \cdots + x_n y_n.$$

**Theorem 2.2** (Verallgemeinerter Satz des Pythagoras [Ant98]). Für orthogonale Vektoren  $u, v$  in einem euklidischen Vektorraum  $V$  gilt

$$\|u + v\|^2 = \|u\|^2 + \|v\|^2.$$

**Definition 2.3** (Orthogonalität [Jän07]). Zwei Elemente  $v, w$  eines euklidischen Vektorraumes  $V$  heißen *orthogonal* (geschrieben  $v \perp w$ ) wenn ihr Skalarprodukt null ist, d.h.

$$v \perp w \iff \langle v, w \rangle = 0.$$

Eine Familie  $(v_1, \dots, v_n)$  in  $V$  heißt *orthogonal* oder *Orthogonalsystem*, wenn

$$v_i \perp v_j \quad \text{für alle } i \neq j.$$

Gilt zusätzlich  $\langle v_i, v_i \rangle = 1$  für alle  $1 \leq i \leq n$ , so spricht man von einem *Orthonormalsystem*.

**Theorem 2.4** (Existenz einer Orthonormalbasis [Fis13]). Jeder endlichdimensionale euklidische Vektorraum besitzt eine Orthonormalbasis.

Der Begriff der Orthogonalität lässt sich auf Matrizen übertragen.

**Definition 2.5** (Orthogonale Matrix [Ant98]). Eine Matrix  $A \in \mathbb{R}^{n \times n}$  heißt *orthogonal*, falls deren Zeilen- und Spaltenvektoren paarweise orthonormal bezüglich des Standardskalarprodukts sind, d.h.

$$A^\top A = \mathbb{1}_n$$

**Definition 2.6** (Orthogonalprojektion [Ant98]). Eine *Orthogonalprojektion* auf einen Untervektorraum  $U$  eines Vektorraumes  $V$  ist eine lineare Abbildung  $P_U: V \rightarrow V$ , die für alle Vektoren  $v \in V$  die beiden Eigenschaften

- (i)  $P_U(v) \in U$  (Projektion)
- (ii)  $\langle P_U(v) - v, u \rangle = 0$  für alle  $u \in U$  (Orthogonalität)

erfüllt.

Mithilfe einer Orthonormalbasis für  $U$  lässt sich aus dieser Definition eine Lösung für die Orthogonalprojektion  $P_U(v)$  herleiten.

**Theorem 2.7** ([Ant98]). Ist  $(u_1, \dots, u_n)$  eine Orthogonalbasis von  $U$ , so gilt für alle  $v \in V$

$$P_U(v) = \sum_{i=1}^n \frac{\langle v, u_i \rangle}{\langle u_i, u_i \rangle} u_i$$

Später werden wir die Orthogonalprojektion in einer anderen Form nutzen. Wir können die Projektion auch als Matrix-Vektor-Produkt auffassen. Verwenden wir das Standardskalarprodukt gilt mit einer Orthogonalbasis  $(u_1, \dots, u_n)$  von  $U$ :

$$P_U(v) = \sum_{i=1}^n \frac{v^\top u_i}{u_i^\top u_i} u_i = \sum_{i=1}^n \frac{u_i u_i^\top}{u_i^\top u_i} v = A A^\top v \quad (2.1)$$

wobei  $\mathbf{A} = \begin{bmatrix} \frac{u_1}{\|u_1\|} & \cdots & \frac{u_n}{\|u_n\|} \end{bmatrix}$ . Die Orthonormalitätsbedingung in Theorem 2.7 kann auch weggelassen werden. Ist  $(u_1, \dots, u_n)$  eine beliebige Basis von  $U$ , so gilt:

$$P_U(v) = \mathbf{A}(\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top v \quad (2.2)$$

Wir nennen  $\mathbf{P} = \mathbf{A}(\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top$  die *orthogonale Projektionsmatrix*. Mithilfe von Theorem 2.2 lässt sich zeigen, dass der orthogonal auf den Unterraum projizierte Vektor den Abstand zwischen dem Ausgangsvektor und dem Unterraum minimiert.

**Theorem 2.8** ([Ant98]). Sei  $U$  ein Unterraum eines euklidischen Vektorraumes  $V$ . Dann ist  $P_U(v)$  die beste Näherung von  $u$  in  $U$ , d.h.

$$\|P_U(v) - v\|^2 \leq \|u - v\|^2 \quad \text{für alle } u \in U$$

## 2.1.2 Matrixzerlegungen

In diesem Abschnitt führen wir zwei wichtige Matrixzerlegungen ein, die auch in vielen Bereichen der Numerik Anwendung finden.

**Definition 2.9** (Eigenwert, Eigenvektor [Ant98]). Sei  $\mathbf{A} \in \mathbb{R}^{n \times n}$ . Ein von Null verschiedener Vektor  $x \in \mathbb{R}^n$  heißt *Eigenvektor* von  $\mathbf{A}$ , falls

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

für einen Skalar  $\lambda \in \mathbb{R}$ . Die Zahl  $\lambda$  heißt *Eigenwert* von  $\mathbf{A}$ .

**Definition 2.10** (Diagonalisierbar [Ant98]). Eine quadratische Matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  heißt *diagonalisierbar*, wenn eine invertierbare Matrix  $\mathbf{V}$  existiert, so dass  $\mathbf{\Lambda} = \mathbf{V}^{-1} \mathbf{A} \mathbf{V}$  Diagonalgestalt hat.

Es gibt verschiedene Kriterien für die Diagonalisierbarkeit von Matrizen. Für unsere spätere Anwendung interessieren wir uns vor allem für die Frage, ob es zu einer gegebenen Matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  eine orthogonale Matrix  $\mathbf{V}$  gibt, die  $\mathbf{A}$  diagonalisiert. Eine derartige Diagonalisierung wird auch als *Hauptachsentransformation* bezeichnet. Dieser Name stammt ursprünglich aus der Theorie der Kegelschnitte. Hierbei ist eine Hauptachsentransformation eine orthogonale Abbildung, welche die Koordinatenachsen in die Richtungen der beiden *Hauptachsen* überführt. Wir wollen uns aber vorerst nicht mit dieser geometrischen Interpretation beschäftigen, sondern mit einem mathematisch äquivalenten, in den Anwendungen aber wichtigeren Problem.

**Theorem 2.11** (Hauptachsentransformation [Jän07]). Sei  $\mathbf{A} \in \mathbb{R}^{n \times n}$  eine symmetrische Matrix. Dann gibt es eine orthogonale Transformation  $\mathbf{V}$ , welche  $\mathbf{A}$  in eine Diagonalmatrix  $\mathbf{\Lambda} := \mathbf{V}^{-1} \mathbf{A} \mathbf{V}$  der Gestalt

$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & & & & \\ & \ddots & & & \\ & & \lambda_1 & & \\ & & & \ddots & \\ & & & & \lambda_r & & \\ & & & & & \ddots & \\ & & & & & & \lambda_r \end{bmatrix}$$

überführt. Hierbei sind  $\lambda_1, \dots, \lambda_r$  die verschiedenen Eigenwerte von  $\mathbf{A}$ .

Zusammenfassend besitzt eine symmetrische Matrix also eine Zerlegung  $\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top$ . Man kann  $\mathbf{V}$  konstruieren, so dass die Spalten genau den Eigenvektoren von  $\mathbf{A}$  entsprechen. Wir werden diese Umformung in späteren Kapiteln unter dem Begriff *Eigenwertzerlegung* (Englisch: *Eigenvalue Decomposition*) verwenden.

Eine eng verwandte, aber vielseitigere Faktorisierung von Matrizen ist die *Singulärwertzerlegung*. Sie ermöglicht eine allgemeine Zerlegung von nicht quadratischen oder nicht symmetrischen Matrizen.

**Theorem 2.12** (Singulärwertzerlegung [SW04]). *Jede Matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$  besitzt eine Singulärwertzerlegung*

$$\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^\top$$

mit orthogonalen Matrizen  $\mathbf{U} \in \mathbb{R}^{m \times m}$  und  $\mathbf{V} \in \mathbb{R}^{n \times n}$ , sowie der Diagonalmatrix  $\mathbf{D} = (\sigma_j \delta_{ij}) \in \mathbb{R}^{m \times n}$ .

**Definition 2.13** (Singulärwert). Die positiven Diagonaleinträge  $\sigma_i > 0$  von  $\mathbf{D}$  werden *Singulärwerte* genannt.

Singulärwerte einer Matrix  $\mathbf{A}$  sind eindeutig bestimmt und stehen durch  $\sigma_i = \sqrt{\lambda_i}$  in einer engen Beziehung mit den Eigenwerten  $\lambda_i$ . Konventionell werden die Singulärwerte von  $\mathbf{D}$  absteigend sortiert, d.h.  $\sigma_1 \geq \dots \geq \sigma_r$ . Geometrisch bedeutet diese Zerlegung, dass sich die Matrix  $\mathbf{A}$  in zwei Drehungen  $\mathbf{U}, \mathbf{V}$  und eine Streckung unterteilen lässt. Dabei korrespondieren die Streckungsfaktoren mit den Einträgen der Diagonalmatrix  $\mathbf{D}$ .

### 2.1.3 Matrix Approximation

In diesem Abschnitt werden wir zwei Approximationsprobleme für Matrizen formulieren, die eine explizite Lösung besitzen. Zunächst führen wir dafür eine Matrixnorm ein, von welcher wir auch später sehr häufig Gebrauch machen werden.

**Definition 2.14** (Frobeniusnorm [SW04]). Für eine Matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$  ist die *Frobeniusnorm* definiert durch

$$\|\mathbf{A}\|_F = \left( \sum_{i=1}^m \sum_{j=1}^n |a_{ij}|^2 \right)^{\frac{1}{2}}.$$

Man zeigt leicht, dass  $\|\mathbf{A}\|_F^2 = \text{tr}(\mathbf{A}^\top \mathbf{A})$  gilt. Eine weitere wichtige Eigenschaft von Matrizen ist der *Rang*.

**Definition 2.15** (Rang [Ant98]). Die Dimension des Zeilen- und des Spaltenraumes einer Matrix  $\mathbf{A}$  heißt *Rang* von  $\mathbf{A}$  und wird mit  $\text{rank}(\mathbf{A})$  bezeichnet.

Wir möchten nun eine Matrix  $\mathbf{A}$  durch eine andere, einfachere Matrix  $\hat{\mathbf{A}}$  mit niedrigerem Rang approximieren. Dieses Problem fällt unter die Kategorie *low rank approximation*, welche eine enge Verbindung zur Hauptkomponentenanalyse aufweist. In Anwendung korrespondiert die Rang-Bedingung mit der Komplexität eines Modells. Mithilfe der Singulärwertzerlegung können wir eine explizite Lösung angeben.

**Theorem 2.16** (Eckart-Young-Mirsky-Theorem [EY36]). *Sei  $\mathbf{A} \in \mathbb{R}^{m \times n}$  mit  $m \leq n$  und*

$$\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^\top$$

eine Singulärwertzerlegung von  $\mathbf{A}$ . Wir partitionieren  $\mathbf{U}$ ,  $\mathbf{D}$  und  $\mathbf{V}$  wie folgt:

$$\mathbf{U} =: [\mathbf{U}_1 \quad \mathbf{U}_2], \quad \mathbf{D} =: \begin{bmatrix} \mathbf{D}_1 & 0 \\ 0 & \mathbf{D}_2 \end{bmatrix}, \quad \mathbf{V} =: [\mathbf{V}_1 \quad \mathbf{V}_2],$$

wobei  $\mathbf{U}_1 \in \mathbb{R}^{m \times r}$ ,  $\mathbf{D}_1 \in \mathbb{R}^{r \times r}$  und  $\mathbf{V}_1 \in \mathbb{R}^{n \times r}$ . Dann löst die abgeschnittene Singulärwertzerlegung (Englisch: *truncated singular value decomposition*)

$$\hat{\mathbf{A}}^* = \mathbf{U}_1 \mathbf{D}_1 \mathbf{V}_1^\top,$$

das Approximationsproblem

$$\min_{\text{rank}(\hat{\mathbf{A}}) \leq r} \|\mathbf{A} - \hat{\mathbf{A}}\|_F = \|\mathbf{A} - \hat{\mathbf{A}}^*\|_F = \sqrt{\sigma_{r+1}^2 + \dots + \sigma_m^2}, \quad (2.3)$$

wobei  $\sigma_i$  die Singulärwerte von  $\mathbf{A}$  sind. Der Minimierer  $\hat{\mathbf{A}}^*$  ist genau dann eindeutig, wenn  $\sigma_{r+1} \neq \sigma_r$ .

Das Eckart-Young-Mirsky-Theorem wird es uns in Abschnitt 3.5 ermöglichen, eine wertvolle Eigenschaft der Hauptkomponentenanalyse zu zeigen.

Ein anderes Approximationsproblem für Matrizen ist das *orthogonale Procrustes Rotationsproblem*. Hierbei sind uns zwei Matrizen  $\mathbf{M}$  und  $\mathbf{N}$  gegeben, welche durch eine orthogonale Transformation ineinander überführt werden sollen. Wieder hilft uns die Singulärwertzerlegung bei der Findung einer Lösung.

**Theorem 2.17** (Procrustes Rotationsproblem [GD+04]). Seien  $\mathbf{M} \in \mathbb{R}^{n \times p}$ ,  $\mathbf{N} \in \mathbb{R}^{n \times k}$  und  $\mathbf{M}^\top \mathbf{N} = \mathbf{U} \mathbf{D} \mathbf{V}^\top$  eine Singulärwertzerlegung. Dann löst

$$\hat{\mathbf{A}} = \mathbf{U} \mathbf{V}^\top$$

das Approximationsproblem

$$\hat{\mathbf{A}} = \arg \min_{\mathbf{A}} \left\| \mathbf{M} - \mathbf{N} \mathbf{A}^\top \right\|_F^2 \quad (2.4)$$

unter der Nebenbedingung, dass  $\mathbf{A}^\top \mathbf{A} = \mathbf{I}_{k \times k}$ .

In Abschnitt 5.1 wird sich (2.4) als Subproblem der dünnbesetzten Hauptkomponentenanalyse herausstellen.

## 2.2 Analysis

In diesem Abschnitt möchten wir die

### 2.2.1 Norm

**Definition 2.18** ([Hie19]). Eine Abbildung  $\|\cdot\| : \mathbb{R}^n \rightarrow \mathbb{R}_0^+$  heißt *Norm*, falls für alle Vektoren  $x, y \in \mathbb{R}^n$  und alle Skalare  $\alpha \in \mathbb{R}$  die folgenden drei Axiome gelten:

- (i)  $\|x\| = 0 \Leftrightarrow x = 0$  (Definitheit)
- (ii)  $\|\alpha \cdot x\| = |\alpha| \cdot \|x\|$  (Homogenität)
- (iii)  $\|x + y\| \leq \|x\| + \|y\|$  (Subadditivität)

**Definition 2.19** ( $\ell_p$ -Norm [SW04]). Auf dem  $\mathbb{R}^n$  sind die  $\ell_p$ -Normen für  $1 \leq p < \infty$  definiert als

$$\|x\|_p := \left( \sum_{i=1}^n |x_i|^p \right)^{\frac{1}{p}} \quad x \in \mathbb{R}^n$$

und für  $p = \infty$  als

$$\|x\|_\infty := \max_{1 \leq i \leq n} |x_i| \quad x \in \mathbb{R}^n.$$

Im Fall  $p = \infty$  spricht man auch von der *Maximumsnorm* und im Fall  $p = 2$  von der *euklidischen Norm*.

Um Verwirrung auszuschließen werden wir im Folgenden von der  $\ell_q$ -Norm sprechen, da wir mit  $p$  die Anzahl an Variablen in einem Modell bezeichnen. Eine weitere wichtige Norm, die wir im Zuge dieser Arbeit verwenden werden ist die  $\ell_0$ -Norm. Diese zählt die von null verschiedenen Einträge eines Vektors und misst somit, ob ein Vektor dünnbesetzt ist.

**Definition 2.20** ( $\ell_0$ -Norm [FR13]). Die sogenannte  $\ell_0$ -Norm ist definiert durch

$$\|x\|_0 := |\{i: x_i \neq 0, \quad 1 \leq i \leq n\}|.$$

Die übliche Schreibweise  $\|x\|_0$  - die Notation  $\|x\|_0^0$  wäre angemessener - entspringt der Beobachtung, dass

$$\|x\|_q^q = \sum_{i=1}^n |x_i|^q \xrightarrow{q \rightarrow 0} \sum_{i=1}^n \mathbb{1}_{\{x_i \neq 0\}} = |\{i: x_i \neq 0, \quad 1 \leq i \leq n\}|$$

Wir werden diese Schreibweise analog für Matrizen anstatt Vektoren verwenden. Dabei wollen wir betonen, dass die  $\ell_0$ -Norm keine wirkliche Norm ist, da die Abbildung nicht homogen ist. Trotzdem ist diese "Norm" in der Theorie der komprimierten Erfassung (Englisch: *compressive sensing*) sehr nützlich. Des Weiteren werden wir die Notation  $\|\cdot\|_q$  gemäß der Definition in 2.19 auch für Werte  $0 < q < 1$  verwenden, obwohl durch diese Abbildung ebenfalls keine Norm gegeben ist.

## 2.3 Generalisierte lineare Modelle

Es existiert eine sehr enge Verbindung zwischen der Hauptkomponentenanalyse, die wir in Kapitel 3 näher kennenlernen werden, und der Regressionsanalyse. Viele der Ideen und Ansätze im folgendem Abschnitt werden wir später gebrauchen und spielen eine maßgebliche Rolle bei der Formulierung der dünnbesetzten Hauptkomponentenanalyse.

### 2.3.1 Grundlagen aus der Statistik

Seien  $(x_1, \dots, x_n)$  Datenpunkte. Es bezeichne  $\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i$  das Stichprobenmittel. Stichprobenvarianz

$$s_x^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x}_n)^2$$

Empirische Kovarianzmatrix!

Sei  $g: \Theta \rightarrow \mathbb{R}$  eine zu schätzende reelle Parameterfunktion in einem statistischem Modell  $(X, \mathcal{A}, \mathcal{P})$  wobei  $\mathcal{P} = \{P_\theta: \theta \in \Theta\}$ . Wir werden nun einige wichtige Grundbegriffe für einen Schätzer  $d: (X, \mathcal{A}) \rightarrow (\mathbb{R}, \mathcal{B}(\mathbb{R}))$  in  $\mathcal{L}^1(\mathcal{P})$  einführen.

**Definition 2.21** (Verzerrung [Rüs14]). Die *Verzerrung* (Englisch: *Bias*) des Schätzers  $d$  bei  $\vartheta$  ist definiert durch

$$\text{Bias}_\vartheta[d] := \mathbb{E}_\vartheta[d] - g(\vartheta).$$

Um verschiedene Schätzer miteinander zu vergleichen bedienen wir uns häufig des *mittleren quadratischen Fehlers*. Dieser gibt an, welche Abweichung zwischen dem Schätzer und dem wahren Parameter zu erwarten ist. Damit bietet sich uns eine Möglichkeit den erwarteten Fehler eines Lernalgorithmus analysieren.

**Definition 2.22** (Mittlerer quadratischer Fehler [Koh05]). Der *mittlere quadratische Fehler* (Englisch: *Mean Squared Error (MSE)*) ist definiert durch

$$\text{MSE}(d, \vartheta) := \mathbb{E}_\vartheta \left[ (d - g(\vartheta))^2 \right].$$

**Theorem 2.23** (Verschiebungssatz [Koh05]). Der *mittlere quadratische Fehler* zerfällt in *Varianz* und *Bias*, d.h.

$$\text{MSE}(d, \vartheta) = \text{Var}_\vartheta[d] + (\text{Bias}_\vartheta[d])^2$$

Für die Bewertung eines Schätzers ist also sowohl Verzerrung als auch Varianz zu berücksichtigen. Leider ist in der Praxis selten möglich, beide Fehlerquellen zeitgleich zu minimieren. Im Bereich des überwachten maschinellen Lernens ist das Problem unter dem *Verzerrung-Varianz-Dilemma* (Englisch: *bias-variance tradeoff*) bekannt. Idealerweise versucht man ein Modell zu wählen, welches sowohl die Gesetzmäßigkeiten in den Trainingsdaten genau erfasst, als sich auch auf ungesehene Testdaten generalisieren lässt. Aufgrund von falschen Annahmen kann es bei einem Lernalgorithmus zu einer hohen Verzerrung kommen. Beziehungen zwischen Eingabe und Ausgabe können nicht geeignet modelliert werden und es kommt zu einem Fehler zwischen System und Modell. Man spricht in diesem Fall von einer *Unteranpassung* (Englisch: *underfitting*). Demgegenüber sind Modelle mit hoher Varianz meist komplexer und ermöglichen eine präzise Darstellung der Trainingsdaten. Dadurch läuft man aber Gefahr, sich dem Rauschen der Daten anzupassen und nicht die Gesetzmäßigkeiten der Trainingsdaten zu erkennen. Wir bezeichnen dieses Phänomen als *Überanpassung* (Englisch: *overfitting*), was in ungenauen Vorhersagen auf Testdaten münden kann.

Die Frage nach einem günstigen Modell liegt also in der Modellkomplexität und es gilt eine Balance zwischen den beiden beschriebenen Extrema zu finden. Diese Idee haben wir in 2.1 veranschaulicht.

### 2.3.2 Lineare Regression

Bei der Regressionsanalyse werden Zusammenhänge zwischen mehreren Merkmalen untersucht. Man versucht eine unabhängige Variable  $Y$  durch eine oder mehrere abhängige Variablen  $X_1, \dots, X_p$  zu erklären. Ein lineares Regressionsmodell hat also die Form

$$f(X) = \beta_0 + \sum_{j=1}^p X_j \beta_j \tag{2.5}$$

wobei  $\beta_j$  die Regressionskoeffizienten sind. Bei der Verwendung dieses Modells nehmen wir an, dass die Regressionsfunktion  $\mathbb{E}(Y|X)$  linear ist bzw. ein lineares Modell eine geeignete Approximation ist [HTF09].

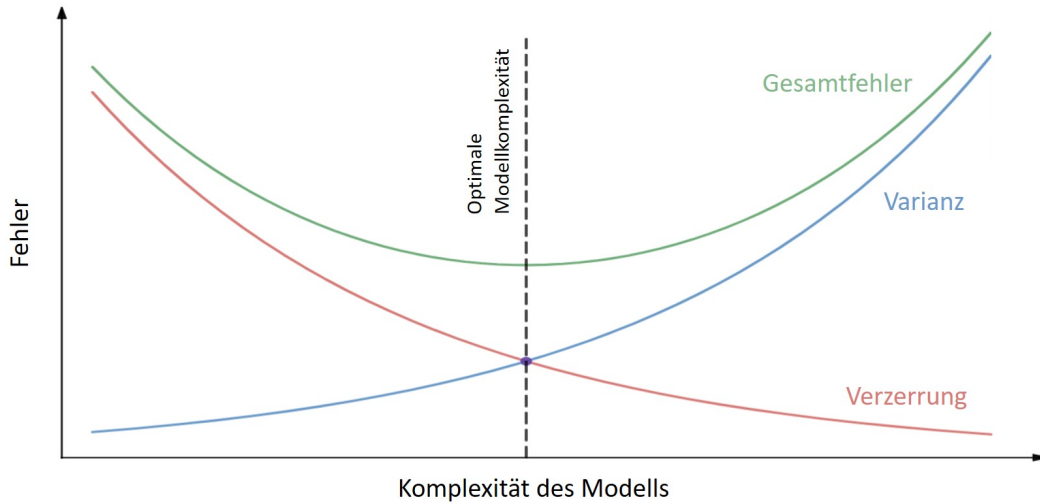


ABBILDUNG 2.1: Verzerrung-Varianz-Dilemma

Typischerweise verfügen wir über eine Menge von Trainingsdaten  $(x_1, y_1), \dots, (x_n, y_n)$ . Jedes  $x_i = (x_{i1}, \dots, x_{ip})^\top$  stellt eine Beobachtung dar, die wir für die Schätzung der Parameter  $\beta_j$  benutzen. Die bekannteste Methode für diesen Zweck ist sicherlich die *Methode der kleinsten Quadraten* (Englisch: *(Ordinary) Least Squares*), in welcher wir  $\beta = (\beta_0, \dots, \beta_p)^\top$  so wählen, dass die Summe der Residuenquadrate (Englisch: *Residual Sum of Squares (RSS)*) minimiert wird. Wir definieren

$$\begin{aligned}
 \text{RSS}(\beta) &= \sum_{i=1}^n (y_i - f(x_i))^2 \\
 &= \sum_{i=1}^n (y_i - \beta_0 - \sum_{j=1}^p x_{ij}\beta_j)^2 \\
 &= (y - \mathbf{X}\beta)^\top (y - \mathbf{X}\beta) \\
 &= \|y - \mathbf{X}\beta\|_2^2
 \end{aligned} \tag{2.6}$$

und das dazugehörige Minimierungsproblem

$$\hat{\beta}^{\text{OLS}} = \arg \min_{\beta} \text{RSS}(\beta) \tag{2.7}$$

wobei  $\mathbf{X} \in \mathbb{R}^{n \times (p+1)}$  die Matrix der  $x_i$  mit einer 1 an erster Stelle ist und  $y = (y_1, \dots, y_n)^\top$ . An dieser Stelle möchten wir erwähnen, dass bei Verwendung dieser Methode keine Aussage über die Gültigkeit des Modells getroffen, sondern lediglich die beste lineare Approximation gefunden wird.

Falls  $\mathbf{X}$  vollen Rang hat zeigt man leicht, dass (2.7) die eindeutige Lösung

$$\hat{\beta}^{\text{OLS}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top y \tag{2.8}$$

besitzt. Die Zielgröße  $y$  ergibt sich dann durch

$$\hat{y} = \mathbf{X} \hat{\beta}^{\text{OLS}} = \mathbf{X} (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top y \tag{2.9}$$



Die Matrix  $\mathbf{P} = \mathbf{X}(\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top$  haben wir bereits in Abschnitt 2.1.1 kennengelernt. Sie projiziert  $y$  orthogonal auf den durch die Spalten von  $\mathbf{X}$  aufgespannten Unterraum. Dies ermöglicht eine geometrische Interpretation der linearen Regression.

Wenn  $\mathbf{X}$  keinen vollen Rang hat, ist die Lösung von (2.7) nicht mehr eindeutig. Dieser Art Probleme ereignen sich häufig in der Bild- und Signalanalyse, bei welcher wir häufig über mehr Variablen als Beobachtungen verfügen. Um ein gewünschtes Verhalten der Regression zu gewährleisten, bestehen verschiedene Möglichkeiten der Filterung oder Regularisierung. In letzterem Fall versehen wir den Regressionsterm mit sog. *Straftermen*, welche eine bedeutende Rolle in den folgenden Kapitel spielen werden. Wir möchten diese mithilfe der linearen Regression einführen. Die Effekte der Strafterme können aber auf andere verallgemeinerte lineare Modelle übertragen werden.

Die von uns eingeführten Strafterme werden vor allem eine Schrumpfung der Regressionskoeffizienten verursachen. Daher möchten wir zunächst motivieren, in welchen Situationen eine derartige Regularisierung hilfreich sein kann.

- Vorhersagegenauigkeit: Besonders in  $p > n$ -Fällen neigt ein lineares Regressionsmodell häufig zu einer Überanpassung an die Trainingsdaten. Um die Vorhersagegenauigkeit für ungesehene Testdaten zu verbessern, kann eine Erhöhung des Bias im Sinne des Verzerrung-Varianz-Dilemmas sinnvoll sein. Dies können wir beispielsweise dadurch erreichen, dass wir die Regressionskoeffizienten verkleinern oder sogar auf 0 setzen.
- Interpretation: Eine hohe Anzahl an Variablen, die in das Modell einfließen erschwert unzweifelhaft eine Interpretation. Daher kann es von Vorteil sein nur einen Teil der Variablen für das Modell auszuwählen. Optimalerweise selektiert man solche, die eine möglichst genaue Vorhersage ermöglichen.

Ein naheliegender Ansatz zur Lösung dieser Probleme wäre es zu versuchen, eine  $k$ -Teilmenge der Variablen zu finden, die eine minimale Summe der Residuenquadrate aufweist. In [HTF09] werden verschiedene Methoden zur exakten und approximativen Berechnung dieser Teilmenge beschrieben. Nicht immer wird die Genauigkeit der Vorhersagen durch Verwendung dieses Ansatzes besser. Dies liegt vor allem daran, dass Variablen für das Modell entweder ausgewählt oder verworfen werden. Daher beschäftigen wir uns nun mit Methoden, die eine kontinuierliche Schrumpfung der Regressionskoeffizienten erlauben.

### 2.3.3 Ridge Regression

Zu diesem Zweck kann die *Tikhonov Regularisierung*, die auch unter dem Namen *Ridge Regression* bekannt ist, genutzt werden. Mithilfe eines *Ridge Strafterms* opfern wir einen Teil des Bias für verbesserte Vorhersagen im Sinne des Verzerrung-Varianz-Dilemmas. Wir formulieren das Ridge Regression Problem in der Lagrange Form

$$\hat{\beta}^{\text{ridge}} = \arg \min_{\beta} \sum_{i=1}^n (y_i - \beta_0 - \sum_{j=1}^p x_{ij} \beta_j)^2 + \lambda_2 \sum_{j=1}^p \beta_j^2 \quad (2.10)$$

wobei  $\lambda_2 \geq 0$  ein Parameter ist, der die Stärke der Schrumpfung der Regressionskoeffizienten kontrolliert. Je größer  $\lambda_2$ , desto stärker ist die Schrumpfung der  $\beta_j$ . Durch die Einführung des  $\ell_2$ -Strafterms garantiert (2.10) auch für  $p > n$  eine eindeutige Lösung. Da  $\beta_0$  nicht im Strafterm vorkommt, schätzen wir zunächst  $\beta_0 = \bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$  und zentrieren die Eingaben

$x_{ij} = x_i - \bar{x}_j$ . Die eindeutige Lösung der zentrierten Version von (2.10) ist dann durch

$$\hat{\beta}^{\text{ridge}} = (\mathbf{X}^\top \mathbf{X} + \lambda_2 \mathbf{I})^{-1} \mathbf{X}^\top \mathbf{y} \quad (2.11)$$

gegeben, wobei  $\beta = (\beta_1, \dots, \beta_p)$  und  $\mathbf{X} \in \mathbb{R}^{n \times p}$  die Matrix der  $x_i$ . Die durch die Ridge Regression erzeugten Koeffizienten sind also um den Faktor  $\frac{1}{1+\lambda_2}$  gegenüber denen der klassischen linearen Regression skaliert. Eine Dünnbesetzung der Koeffizienten wird also erst für  $\lambda_2 \rightarrow \infty$  erreicht.

### 2.3.4 LASSO

Um eine bessere Interpretation des Modells zu ermöglichen, versucht man bei der *Lasso* Regression eine Lösung zu finden, bei welcher viele Koeffizienten gleich Null sind. Das Lasso wurde erstmals von Tibshirani in [Tib96] eingeführt und ist in der Signalanalyse unter dem Namen *Basis Pursuit* [CDS98] bekannt. Mathematisch gesehen erreichen wir eine Dünnbesetzung durch Einbettung eines  $\ell_1$ -Strafterms.

$$\hat{\beta}^{\text{lasso}} = \arg \min_{\beta} \sum_{i=1}^n (y_i - \beta_0 - \sum_{j=1}^p x_{ij} \beta_j)^2 + \lambda_1 \sum_{j=1}^p |\beta_j| \quad (2.12)$$

Es wird also im Vergleich zu (2.10) lediglich die  $\ell_2$ -Norm durch eine  $\ell_1$ -Norm ausgetauscht. Bevor wir uns mit der Lösung dieses Problems beschäftigen, möchten wir erklären, warum die  $\ell_1$ -Norm eine Dünnbesetzung hervorruft. Zunächst geben wir eine geometrische Erklärung, welche in Abbildung 2.2 zu sehen ist. Dort sind die  $\ell_1$ - und  $\ell_2$ -Beschränkungen, sowie die Höhenlinien der RSS-Funktion in zwei Dimensionen aufgezeichnet. Die optimalen Koeffizienten von Ridge und Lasso Regression ergeben sich nun aus dem Schnittpunkt der Höhenlinien mit der Begrenzung der jeweiligen Norm. Im Falle der  $\ell_1$ -Norm ist dieser Schnittpunkt mit einer hohen Wahrscheinlichkeit an eine der Ecken, so dass einer der beiden Koeffizienten auf Null gesetzt wird. Im Gegensatz dazu ist die Wahrscheinlichkeit bei einer  $\ell_2$ -Begrenzung sehr gering, da keine Ecken vorhanden sind. Somit kommt jeder Randpunkt der Begrenzung als Schnittpunkt in Frage und es wird keine Dünnbesetzung, sondern lediglich eine kontinuierliche Schrumpfung der Koeffizienten hervorgerufen. Dieser Effekt verstärkt sich in höheren Dimensionen.

An dieser Stelle kann man auf den Gedanken kommen, andere Strafterme zu verwenden, welche bei geometrischer Betrachtung die Wahrscheinlichkeit erhöhen eine Dünnbesetzung der Koeffizienten hervorzurufen. So kann man zum Beispiel die  $\ell_q$ -Normen als Strafterm für Werte  $q < 1$  in Betracht ziehen. In Abbildung 2.3 sind die Begrenzungen der  $\ell_q$ -Normen für verschiedene Werte von  $q$  eingezeichnet. Für  $q \rightarrow 0$  entstehen sternförmige Höhenlinien, welche immer weiter zum Ursprung gedrückt werden. Somit wird es immer wahrscheinlicher, dass die Höhenlinien der RSS-Funktion eine Ecke treffen und wir dünnbesetzte Koeffizienten erhalten. Daher kann man folgendes Berechnungsproblem definieren

$$\hat{\beta}^{\text{sparse}} = \arg \min_{\beta} \sum_{i=1}^n (y_i - \beta_0 - \sum_{j=1}^p x_{ij} \beta_j)^2 + \lambda_q \sum_{j=1}^p |\beta_j|^q \quad (2.13)$$

Leider ist dies nur in der Theorie ein guter Ansatz. Das Problem liegt nicht im Effekt der verschiedenen Strafterme, sondern in der Berechnung. Für  $q < 1$  ist (2.13) ein nicht-konvexes Optimierungsproblem, da  $\|\cdot\|_q$  keine Norm gemäß Definition 2.18 ist. Im Extremfall der  $\ell_0$ -Norm wird (2.13) sogar NP-schwer [FR13]. Somit besteht keine effiziente Methode zur Berechnung von  $\hat{\beta}^{\text{sparse}}$  zur Verfügung. Der Wert  $q = 1$  ist eine Art Kompromisslösung, die

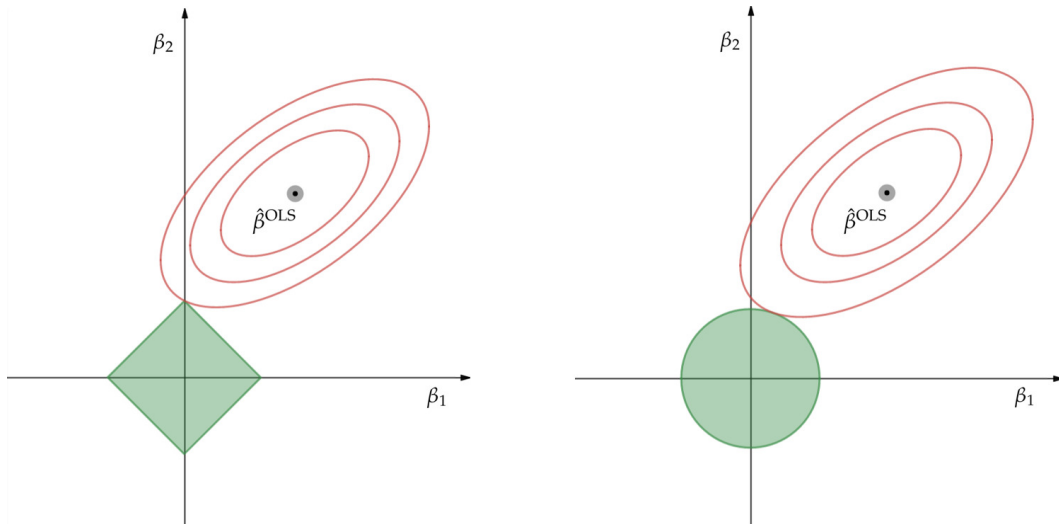


ABBILDUNG 2.2: Die Abbildung zeigt die Beschränkungen der  $\ell_1$ -Norm (links) und der  $\ell_2$ -Norm (rechts) zusammen mit den Höhenlinien der RSS-Funktion im  $\mathbb{R}^2$ . Verdeutlicht wird hier die geometrische Findung von  $\hat{\beta}^{\text{lasso}}$  (links) und  $\hat{\beta}^{\text{ridge}}$  (rechts). (Abbildung basiert auf [HTF09])

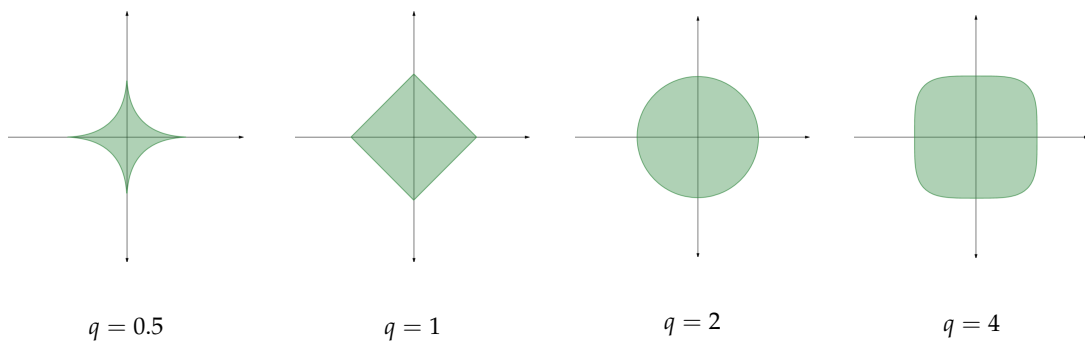


ABBILDUNG 2.3: Die Abbildung zeigt die Begrenzungen der  $\ell_q$ -Norm im  $\mathbb{R}^2$  für verschiedene Werte von  $q$ , also die Mengen  $\{x \in \mathbb{R}^2: \|x\|_q \leq c\}$ . (Abbildung basiert auf [HTF09])

einerseits effizient zu berechnen ist und andererseits noch immer eine dünnbesetzte Lösung liefert.

Um eine mathematisch gründliche Erklärung für die Dünnbesetzung zu liefern, wenden wir uns der Lösung von (2.12) zu. Falls die Spalten von  $\mathbf{X}$  orthonormal sind, ergibt sich eine explizite Lösung

$$\hat{\beta}_j^{\text{lasso}} = \text{sign}(\hat{\beta}_j^{\text{OLS}}) \left( \left| \hat{\beta}_j^{\text{OLS}} \right| - \frac{\lambda_1}{2} \right)_+ \quad (2.14)$$

wobei  $(\cdot)_+ = \max(\cdot, 0)$  ist. Der Beweis kann in [Mur12] nachgelesen werden. Die Lösung ist also durch den sog. *soft thresholding operator* gegeben, welcher durch

$$\text{soft}_\delta(x) = \text{sign}(x)(|x| - \delta)_+ \quad (2.15)$$

definiert wird. Dieser ist in Abbildung 2.4 im Gegensatz zum hard-thresholding Operator dargestellt. Nun sind wir auch in der Lage zu verstehen, warum Tibshirani [Tib96] den Begriff Lasso eingeführt hat, welcher für *Least absolute selection and shrinkage operator* steht. Es

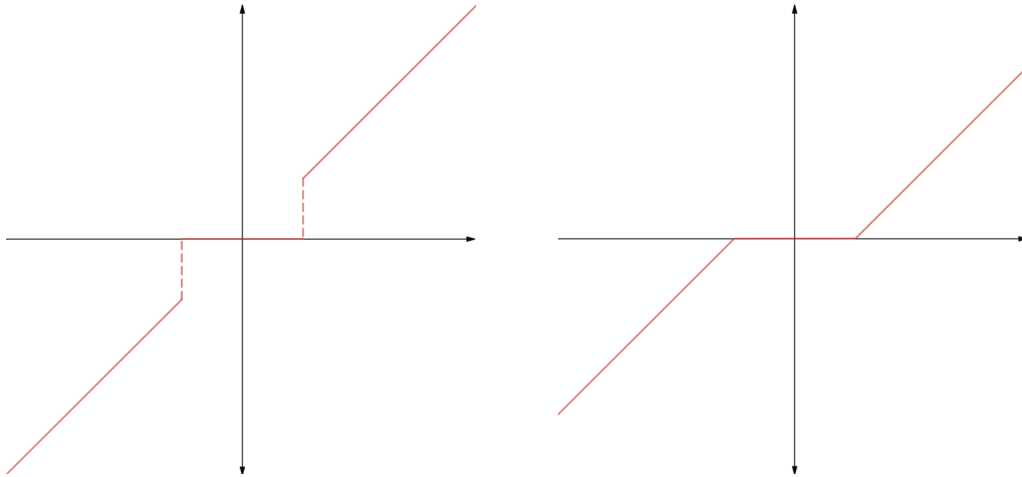


ABBILDUNG 2.4: Die Abbildung zeigt die beiden Operationen soft und hard thresholding

werden zunächst alle Koeffizienten in  $\hat{\beta}^{\text{OLS}}$ , die kleiner als  $\frac{\lambda_1}{2}$  sind, auf 0 gesetzt und die Verbliebenden anschließend im Betrag um  $\frac{\lambda_1}{2}$  geschrumpft.

Für den allgemeinen Fall wird (2.12) mithilfe von Näherungsverfahren gelöst. Da  $\|\beta\|_1$  nicht differenzierbar ist wenn  $\beta_j = 0$ , sind wir mit einem nicht glattem Optimierungsproblem konfrontiert. Seit der Problemformulierung wurde eine Vielzahl an Algorithmen entwickelt bzw. adaptiert, die eine numerische Lösung liefern. Dazu gehören Least-angle Regression (LARS) [Tib+04], Koordinaten-Abstiegsverfahren [FHT10], Subdifferential-Methoden und Näherungs-Gradientenverfahren [Yan+13; Van19]. Letztere sind eine natürliche Erweiterung von Gradientenverfahren falls die Zielfunktion nicht differenzierbar ist. In Abschnitt 2.3.5, werden wir auf ein Koordinaten-Abstiegsverfahren näher eingehen.

Es stellt sich heraus, dass das Lasso zwei wesentliche Nachteile besitzt. Falls es im Datensatz Gruppen stark korrelierter Variablen gibt, tendiert die Methode dazu nur eine Variable je Gruppe statt die Gruppe als Ganzes auszuwählen. Zum Beispiel bei der Suche nach Genen, welche mit einer bestimmten Krankheit verbunden sind, ist man aber daran interessiert, alle assoziierten Koeffizienten zu finden. Darüber hinaus führt dieser Effekt zu verschlechterten Vorhersagen. Um diesem Problem zu entgegnen kann man das sog. *Group Lasso* verwenden [YL06], bei welcher man zuvor Gruppen im Datensatz festlegen kann. Der im Zuge dieser Arbeit aber wichtigere Nachteil ist, dass das Lasso für  $p > n$  Datensätze maximal  $n$  Variablen selektieren kann. Da  $\mathbf{X}$  in diesem Fall maximal Rang  $n$  hat, können wir  $y$  mithilfe von  $n$  Variablen perfekt vorhersagen. Das Lasso wählt dann die  $n$  Variablen, welche  $\lambda_1 \|\beta\|_1$  minimieren. Für moderne Datensätze mit  $p \gg n$  ist es aber oft nicht ausreichend, nur  $n$  von null verschiedenen Koeffizienten zuzulassen.

### 2.3.5 Elastic Net

Damit im Fall  $p > n$  mehr als  $n$  Variablen selektiert werden können, betrachten wir eine Kombination von Lasso und Ridge Regression. Durch die Einbettung einer  $\ell_1$  und  $\ell_2$ -Norm erhalten wir das sog. *Elastic Net* [ZH05].

$$\hat{\beta}^{\text{en}} = \arg \min_{\beta} \|y - \mathbf{X}\beta\|_2^2 + \lambda_2 \|\beta\|_2^2 + \lambda_{1,j} \|\beta\|_1 \quad (2.16)$$

Wie zuvor wird durch den  $\ell_1$ -Strafterm ein dünnbesetztes Modell generiert. Dagegen fördert der  $\ell_2$ -Strafterm den Gruppeneffekt, stabilisiert den  $\ell_1$  Regularisierungspfad und lässt eine beliebige Anzahl zu selektierender Variablen zu. Um eine doppelte Schrumpfung der Koeffizienten zu vermeiden, kann das Elastic Net mit dem Faktor  $(1 + \lambda_2)$  korrigiert werden.

Ähnlich wie bei der Lasso Regression kann nur dann eine explizite Lösung von (2.16) angegeben werden, wenn die Spalten von  $\mathbf{X}$  orthonormal sind. Daher wurden verschiedene Näherungsverfahren für eine numerische Lösung vorgeschlagen. Dazu gehören beispielsweise LARS-EN [ZH05], welches auf dem LARS Algorithmus für das Lasso basiert, und ein Koordinatenabstiegsverfahren [FHT10]. Letzteres werden wir in unserer Implementierung in Kapitel 5 für ein Subproblem nutzen.

<https://stats.stackexchange.com/questions/123672/coordinate-descent-soft-thresholding-update-operator-for-lasso>

We propose an efficient algorithm called LARS-EN to solve the elastic net efficiently, which is based on the recently proposed algorithm LARS of Efron et al. (2004). They proved that, starting from zero, the lasso solution paths grow piecewise linearly in a predictable way. They proposed a new algorithm called LARS to solve the entire lasso solution path efficiently by using the same order of computations as a single OLS fit

Die Implementierung des Elastic Nets in scikit-learn [Ped+11] beruht auf einer anderen, aber sehr ähnlichen mathematischen Formulierung

$$\hat{\beta}^{\text{en}} = \arg \min_{\beta} \frac{1}{2n} \|y - \mathbf{X}\beta\|_2^2 + \alpha\gamma \|\beta\|_1 + \frac{1}{2}\alpha(1 - \gamma) \|\beta\|_2^2 \quad (2.17)$$

wobei  $n$  die Anzahl an Beobachtungen im Datensatz und  $\gamma \in [0, 1]$  das Verhältnis zwischen  $\ell_1$ - und  $\ell_2$ -Norm ist. Mit  $\gamma = 0$  reduziert sich (2.17) auf Ridge und für  $\gamma = 1$  auf die Lasso Regression. Somit können wir durch  $\gamma$  das Verhältnis der beiden Regressionen kontrollieren und durch  $\alpha$  die Stärke der Bestrafung. Setzt man

$$\alpha = \frac{2\lambda_2 + \lambda_1}{2n} \quad \text{und} \quad \gamma = \frac{\lambda_1}{2\lambda_2 + \lambda_1} \quad (2.18)$$

entspricht (2.17) unser ursprünglich formuliertes Problem. An dieser Stelle möchten wir anmerken, dass bei dieser Implementierung nicht die Möglichkeit besteht, die Koeffizienten unterschiedlich zu bestrafen im Gegensatz zu (2.16).

### 2.3.6 Vergleich der Regressionsmethoden

Zur Veranschaulichung der oben eingeführten Methoden werden wir diese nun anwenden. Dabei greifen wir auf einen durch scikit-learn [Ped+11] bereitgestellten Datensatz, der erstmals durch [Tib+04] öffentlich gemacht worden ist, zurück. In diesem wurden für  $n = 442$  Diabetes Patienten  $p = 10$  verschiedene Variablen gemessen. Dazu gehören Alter (AGE), Geschlecht (SEX), Body Mass Index (BMI), Blutdruck (BP) und verschiedene Blutproben (Serum Measurements). Die Zielgröße  $y$  enthält Werte für den Krankheitsfortschritt ein Jahr nach Behandlungsbeginn. Ein Ausschnitt des Datensatzes befindet sich in Tabelle 2.1.

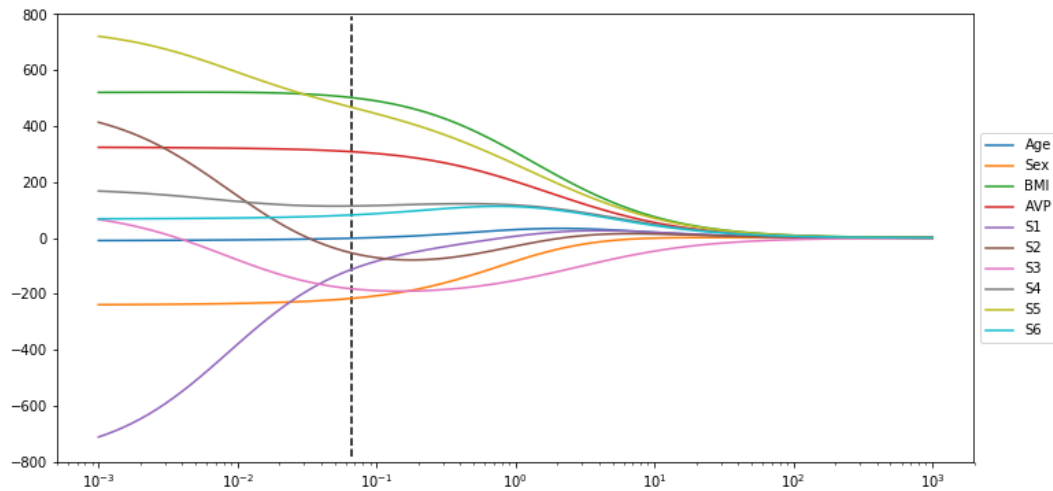
In Abbildung 2.5 sehen wir die Ergebnisse der verschiedenen Regressionsmethoden. Anhand von 2.5 können wir die Schrumpfung der Regressionskoeffizienten mit Veränderung der Regularisierungsparameter  $\lambda_1$ ,  $\lambda_2$  bzw.  $\alpha$  visuell nachvollziehen. Deutlich zu erkennen

Patient	AGE	SEX	BMI	BP	...	Serum Measurements					...	Response
	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10		y
1	59	2	32.1	101	157	93.2	38	4	4.9	87		151
2	48	1	21.6	87	183	103.2	70	3	3.9	69		75
3	72	2	30.5	93	156	93.6	41	4	4.7	85		141
4	24	1	25.3	84	198	131.4	40	5	4.9	89		206
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮		⋮
441	36	1	30.0	95	201	125.2	42	5	5.1	85		220
442	36	1	19.6	71	250	133.2	97	3	4.6	92		57

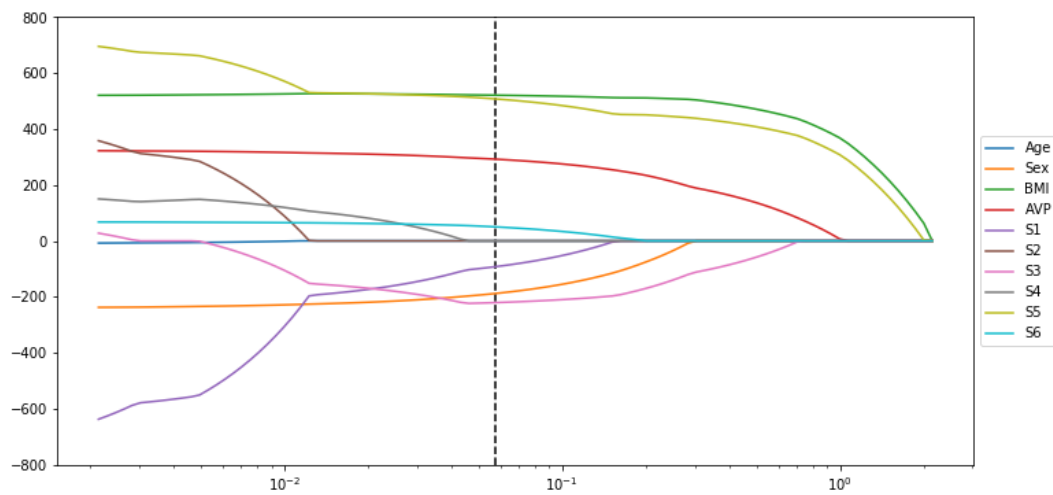
TABELLE 2.1: Überblick über den Diabetes Datensatz aus [Tib+]

ist, dass bei der Ridge Regression eine kontinuierliche Schrumpfung der Koeffizienten stattfindet, d.h. sie werden erst für  $\lambda_2 \rightarrow \infty$  auf Null gesetzt. Im Gegensatz dazu erkennt man bei der Lasso Regression, dass schon für kleine  $\lambda_1$  einzelne Koeffizienten auf Null gesetzt werden und wir eine Dünnbesetzung erhalten. Das Elastic Net bildet eine natürliche Brücke zwischen Ridge und Lasso Regression. Einerseits können wir sehen, dass einzelne Koeffizienten auf Null gesetzt und andererseits, dass die Regularisierungspfade stabilisiert werden.

Für die Wahl von  $\lambda_1$ ,  $\lambda_2$  bzw.  $\alpha$  kann ein  $k$ -faches Kreuzvalidierungsverfahren eingesetzt werden. Hierbei wird der Datensatz in  $k$  möglichst gleich große Teilmengen  $T_1, \dots, T_k$  zerlegt. Anschließend werden  $k$  Durchläufe gestartet, wobei die jeweils  $i$ -te Teilmenge  $T_i$  als Testmenge und die verbleibenden  $k - 1$  Teilmengen  $\{T_1, \dots, T_k\} \setminus \{T_i\}$  als Trainingsmenge verwendet werden. Mittelt man die mittleren quadratischen Fehler der Testmengen, erhält man ein Maß für den Gesamtfehler. Man wählt dann den Regularisierungsparameter mit minimalem Gesamtfehler, welcher in Abbildung 2.5 zu sehen ist.



(A) Ridge Regression



(B) Lasso Regression

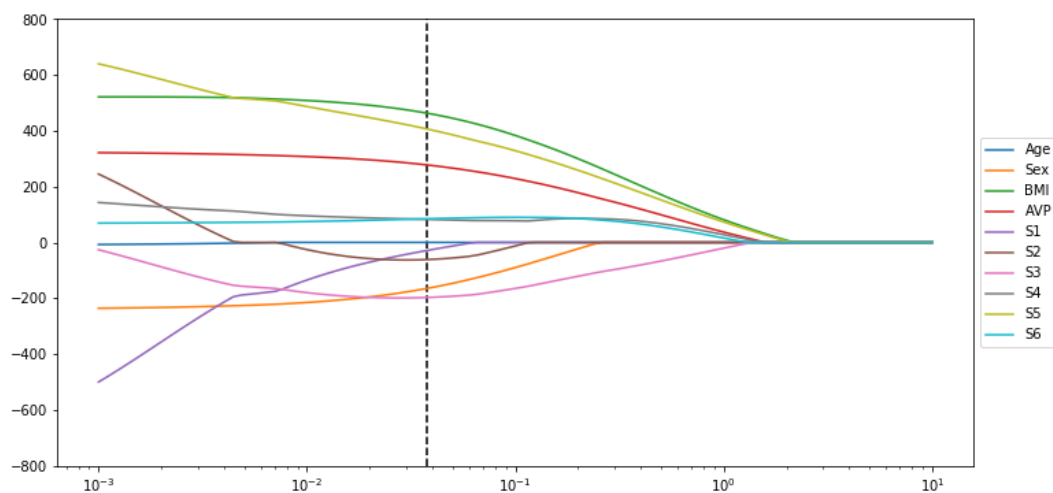
(C) Elastic Net,  $\gamma = 0.99$ 

ABBILDUNG 2.5: Jede Linie stellt den Wert (y-Achse) des jeweiligen Regressionskoeffizienten mit Veränderung der Regularisierungsparameter  $\lambda_2$ ,  $\lambda_1$  bzw.  $\alpha$  (x-Achse) dar. Zu sehen ist die unterschiedliche Art der Schrumpfung für die verschiedenen Strafterme. Die vertikalen Linien entsprechen dem Wert des jeweiligen Parameters, der durch ein 10-faches Kreuzvalidierungsverfahren bestimmt worden ist.





## Kapitel 3

# Hauptkomponentenanalyse

To Do: Kovarianzmatrix / Stichprobenkovarianzmatrix einheitlich! Begriffe wie samples, PCA, oder features erklären, EIGENVALUE = VARIANCE

Die Hauptkomponentenanalyse (Englisch: Principal Component Analysis (PCA)) ist ein weitverbreitetes multivariates statistisches Verfahren zur Dimensionsreduktion. Allgemein zielen derart Verfahren darauf ab, die in einem Datensatz enthaltene Zahl an Variablen zu verringern, ohne dabei die darin enthaltene Information zu verlieren. (Dies mag zunächst widersprüchlich erscheinen, jedoch enthalten Datensätze aufgrund von Korrelationen oft redundante Informationen.) Durch Verringerung der Dimension können umfangreiche Datensätze strukturiert, veranschaulicht und vereinfacht werden. Damit ist das Verfahren Teil der explorativen Statistik, welche Datensätze hinsichtlich ihrer Zusammenhänge analysiert. Die sich ergebende Struktur kann für weitere Analysezwecke ausgenutzt werden.

Aus diesem Grund hat die Hauptkomponentenanalyse in vielen Bereichen erfolgreich Anwendung gefunden. So kann es in der Bildverarbeitung beispielsweise zur Rauschunterdrückung [BSP12] oder zur Gesichtserkennung [Tai+17] genutzt werden. Um Bilder für solch ein Verfahren nutzbar zu machen, werden einzelne Pixel oder patches, also lokale Gruppierungen von Pixeln, eines Bildes als Variable interpretiert. Ein weitere Anwendung in diesem Bereich ist die Erkennung handgeschriebener Zahlen, welche zur automatischen Sortierung von Briefen nach Postleitzahl genutzt werden kann [HTF09]. An diesem Beispiel lässt es sich besonders gut verdeutlichen, was es heißt, Zusammenhänge zu analysieren und Strukturen auf Daten zu finden. Bei Anwendung einer Dimensionsreduktion auf Bilder handgeschriebener Zahlen hofft man, dass sich zehn verschiedene Gruppierungen (Englisch: Cluster) ergeben, die für die Ziffern 0 bis 9 stehen. Idealerweise gehören alle Datenpunkte im selben Cluster zur selben Ziffer. Außerdem korrespondieren nahe beieinanderliegende Cluster mit Ziffern, die ähnlich aussehen. (Bild?)

Das mathematische Problem der Hauptkomponentenanalyse kann auf verschiedene Weisen beschrieben werden. Zunächst wollen wir es so konstruieren, dass die Idee des minimalen Informationsverlust im Vordergrund steht. Anschließend werden wir das Problem auf eine Singulärwertzerlegung zurückführen, die auch zur effizienten Implementierung genutzt wird. Des Weiteren werden wir die Verbindung zwischen der Hauptkomponenten- und der Regressionsanalyse betrachten und die geometrische Interpretation weiter verdeutlichen. Zu Schluss runden wir das Kapitel mit einigen Erweiterungen sowie theoretischen Aussagen ab.

### 3.1 Konstruktion

Gegeben sei ein Datensatz mit  $n$  Beobachtungen und  $p$  Variablen. Die zentrale Idee der Hauptkomponentenanalyse besteht darin, die  $p$  bestehenden Variablen in  $k$  neue, unkorrelierte Variablen zu überführen. Um eine Reduktion der Dimension, also  $k < p$  zu erreichen, müssen die bestehenden Variablen *zusammengefasst* werden. Idealerweise sollte bei diesem Prozess möglichst wenig Information verloren gehen. Als Maß für den Informationsgehalt der Daten wird hierbei die Varianz verwendet. Das heißt, je größer die Varianz einer Variable, desto mehr Information birgt sie und desto *wichtiger* ist sie. Bei der Erkennung von Unterschieden und Strukturen sind Variablen mit niedriger Varianz nicht von Nutzen.

Um die Dimension zu reduzieren könnte man einfach nach Eigenschaften größter Varianz suchen und alle Variablen unterhalb eines festgelegten Grenzwertes verwerfen. Dieses Vorgehen fällt allgemein unter die Kategorie *feature selection*. Sowohl die Hauptkomponentenanalyse als auch viele weitere Dimensionsreduktionsverfahren verwenden allerdings ein anderes Prinzip. Anstatt Eigenschaften mit hoher Varianz auszuwählen, konstruiert man neue Variablen, die sich aus den Bestehenden zusammensetzen. Variablen mit hoher Varianz werden in der Konstruktion einen größeren Beitrag spielen als solche mit niedriger Varianz. Dieser Ansatz ist der Kategorie *feature extraction* zuzuordnen.

Um dieses Prinzip zu veranschaulichen, wenden wir uns einem simplem Beispiel zu. Gegeben seien simulierte Daten, welche Gewicht und Größe zu 1000 Personen beinhalten. Bei Betrachtung der Abbildung 3.1a fällt schnell auf, dass die beiden Variablen positiv korreliert sind, d.h. prinzipiell erkennt man folgende Tendenz: Je größer eine Person, desto schwerer ist sie. Somit können wir einen Großteil an Information in einer neuen Variable zusammenfassen, die sich aus einer Linearkombination von Gewicht und Größe ergibt. Die Koeffizienten der Linearkombination ergeben sich aus der ersten Hauptachse, welche in Richtung größter Varianz zeigt. Projizieren wir unsere Daten auf die erste Hauptachse erhalten wir eine eindimensionale Darstellung. Somit sind Personen, die ähnliches Gewicht oder Größe haben auch im transformierten Raum nahe beieinander. Nach Transformation können wir in diesem Beispiel noch immer knapp 90% der Varianz des ursprünglichen Datensatzes erklären.

#### Vorverarbeitung der Daten

Bevor wir die Hauptkomponentenanalyse auf einen Datensatz anwenden, gibt es einen wichtigen Bearbeitungsschritt zu beachten. Wenn eine Variable weniger variiert als eine Andere aufgrund der verwendeten Einheit oder Skala kann dies zu ungewollten Ergebnissen führen. (REF?) Ohne eine Vorbehandlung der Daten hat so im obigen Beispiel eine Änderung von 1cm die gleiche Bedeutung wie eine Änderung von 1kg. Daher werden die Daten häufig einem Vorverarbeitungsschritt (Englisch: *preprocessing*) unterzogen. Ein zu diesem Zweck oft verwendetes Verfahren ist die Standardisierung oder auch z-Transformation genannt. Hierbei werden die Variablen  $X_i$  zentriert und anschließend auf Einheits-Varianz gebracht. Dies wird erreicht, indem man  $X_i$  durch  $\frac{X_i - E[X_i]}{\sqrt{\text{Var}[X_i]}}$  ersetzt. Mathematisch gesehen wendet man das Verfahren somit anstatt der Stichprobenkovarianzmatrix auf die Stichprobenkorrelationsmatrix an.

#### 3.1.1 Problemformulierung als Varianzmaximierung

Wir wollen nun die Intuition des minimalen Informationsverlust mathematisch beschreiben. Gegeben sei dazu eine Matrix  $\mathbf{X} \in \mathbb{R}^{n \times p}$ , wobei  $n$  die Anzahl der Beobachtungen und  $p$  die Anzahl der Variablen ist. Für eine simplere Darstellung nehmen wir im Folgenden an, dass

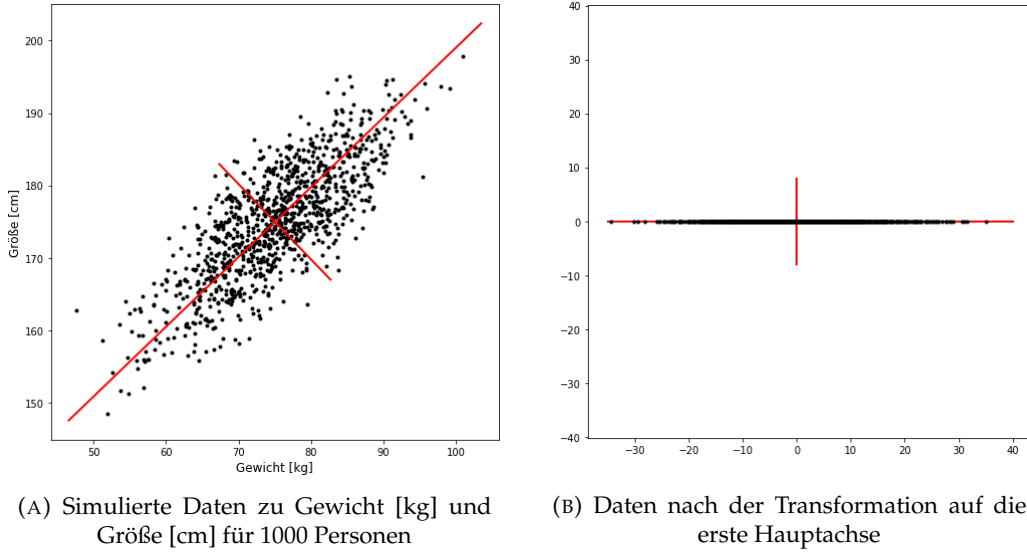


ABBILDUNG 3.1: Die Abbildung zeigt das Ergebnis einer Hauptkomponentenanalyse auf simulierten Daten. Die roten Linien stellen die beiden Hauptachsen, also die Richtungen größter Varianz des Datensatzes dar. Diese sind nach Konstruktion orthogonal.

die Variablen zuvor zentriert wurden. Aufgabe der Hauptkomponentenanalyse ist es nun sukzessive Richtungen größter Varianz zu finden, die sog. *Hauptachsen*. Die Koeffizienten spiegeln dabei den Beitrag jeder einzelnen Variable zur Hauptachse wider. Anschließend werden die *Hauptkomponenten* definiert, welche die Darstellung der Daten bezüglich der neuen Hauptachsen sind. Wir erhalten die erste Hauptachse, indem wir die Varianz von  $Z_1 = \mathbf{X}v_1$ , der ersten Hauptkomponente, maximieren, d.h.

$$v_1 = \arg \max_{\|v\|_2=1} v^T \Sigma v \quad (3.1)$$

wobei  $\Sigma = \frac{\mathbf{X}^T \mathbf{X}}{n-1}$  die Stichprobenkovarianzmatrix ist. Die restlichen Hauptachsen können nun sukzessive definiert werden

$$\begin{aligned} v_{i+1} &= \arg \max_{\|v\|=1} v^T \Sigma v \\ v_{i+1}^T v_l &= 0 \quad \forall 1 \leq l \leq i \end{aligned} \quad (3.2)$$

Man sucht also unter den Richtungen, die orthogonal zu allen bisherigen Hauptachsen sind, diejenige, die die Varianz maximiert. Durch Projektion der Daten  $Z_i = \mathbf{X}v_i$  erhält man dann die Hauptkomponenten [VMS16].

Aufgrund der schrittweisen Konstruktion gibt es eine natürliche Ordnung der Hauptkomponenten. Da keine Restriktion an die erste Hauptachse gestellt wird, erklärt die erste Hauptkomponente den größten Teil der Varianz des Datensatzes. Weitere Hauptachsen müssen orthogonal zu den Vorherigen sein und können somit nur einen geringeren Anteil erklären. Ab einem gewissen Punkt erhalten wir durch Berechnung einer weiteren Hauptkomponente also nur geringfügig mehr Information über den Datensatz. Es gilt einen Punkt der Balance zwischen erklärter Varianz und Modellkomplexität zu finden. Mit dieser Fragestellung werden wir uns weiter in Abschnitt 3.2 beschäftigen.

Für die Maximierungsprobleme (3.1) und (3.2) existiert eine erstaunlich einfache Lösung. Leiten wir (3.1) in der Lagrange-Form  $L(v, \lambda) = v^T \Sigma v + \lambda(1 - v^T v)$  nach  $v$  ab, erhalten wir

die notwendige Bedingung

$$\frac{\partial L(v, \lambda)}{\partial v} = 2\Sigma v - 2\lambda v = 0$$

und somit den stationären Punkt  $\Sigma v_1 = \lambda_1 v_1$ . Das bedeutet, dass die erste Hauptachse genau dem Eigenvektor des größten Eigenwertes  $\lambda_1$  der Stichprobenkovarianzmatrix  $\Sigma$  entspricht. Durch Linksmultiplikation mit  $v_1^T$  sehen wir, dass durch  $v_1^T \Sigma v_1 = \lambda_1$  die Varianz der ersten Hauptkomponente gegeben ist. Analog zeigt man, dass auch die folgenden Hauptachsen, die durch (3.2) definiert sind, den Eigenvektoren von  $\Sigma$  entsprechen [Bis06].

Daher können wir anstatt sukzessiver Berechnung einzelner Hauptachsen die Matrix  $\Sigma$  direkt diagonalisieren. Aufgrund der Symmetrie von  $\Sigma$  können wir eine Eigenwertzerlegung wie in Abschnitt 2.1.2 angeben:

$$\Sigma = \frac{1}{n-1} \mathbf{V} \mathbf{\Lambda} \mathbf{V}^T$$

wobei  $\mathbf{\Lambda}$  eine Diagonalmatrix mit Eigenwerten  $\lambda_i$  und  $\mathbf{V}$  die Matrix der Eigenvektoren ist. Somit können die Hauptachsen direkt aus  $\mathbf{V}$  abgelesen werden. Die Projektion der Daten auf die Hauptachsen wird dann wie zuvor durch Multiplikation der Beobachtungen mit den Eigenvektoren erreicht.

$$\mathbf{Z} = \mathbf{X} \mathbf{V}$$

Die  $i$ -te Spalte in  $\mathbf{Z}$  entspricht also der  $i$ -ten Hauptkomponente und die einzelnen Beobachtungen bezüglich der neuen Darstellung sind die Zeilen von  $\mathbf{Z}$ .

Es gibt einen engen Zusammenhang zwischen der Eigenwertzerlegung von  $\mathbf{X}^T \mathbf{X}$  und der Singulärwertzerlegung von  $\mathbf{X}$ . Diese Beziehung können wir nutzen, um die Lösung noch einfacher zu gestalten. Eine Singulärwertzerlegung ergibt

$$\mathbf{X} = \mathbf{U} \mathbf{D} \mathbf{V}^T$$

wobei  $\mathbf{D}$  die Diagonalmatrix der Singulärwerte,  $\mathbf{U}$  eine orthogonale  $n \times n$  und  $\mathbf{V}$  eine orthogonale  $p \times p$  Matrix ist. Nun sieht man aufgrund der Orthogonalität von  $\mathbf{U}$ , dass

$$\Sigma = \frac{1}{n-1} \mathbf{X}^T \mathbf{X} = \frac{1}{n-1} \mathbf{V} \mathbf{D} \mathbf{U}^T \mathbf{U} \mathbf{D} \mathbf{V}^T = \frac{1}{n-1} \mathbf{V} \mathbf{D}^2 \mathbf{V}^T.$$

Die Singulärwerte stehen also durch  $d_i^2 = \lambda_i$  in Beziehung mit den Eigenvektoren von  $\Sigma$ .

Zusammengefasst können also alle relevanten Ergebnisse einer Hauptkomponentenanalyse mithilfe einer einzelnen Singulärwertzerlegung von  $\mathbf{X}$  angegeben werden. Die Hauptachsen entsprechen den Eigenvektoren in  $\mathbf{V}$ , die Hauptkomponenten der Matrix  $\mathbf{Z} = \mathbf{X} \mathbf{V} = \mathbf{U} \mathbf{D}$  und die zugehörigen Varianzen sind durch  $d_i^2$  gegeben. Für die Berechnung einer solchen Zerlegung stehen äußerst effiziente Verfahren zur Verfügung, welche sowohl den  $n < p$  als auch den  $p < n$  Fall schnell in  $\mathcal{O}(np \cdot \min\{n, p\})$  lösen können.

(So können wir eine Hauptkomponentenanalyse durchführen. In der Praxis wird meist nur die kompakte Singulärwertzerlegung, die nur von Null verschiedene Singulärwerte berücksichtigt, berechnet. Die Anzahl ist gegeben durch den Rang von  $\mathbf{X}$ . Es sei  $k \leq \min\{n, p\}$  der Rang von  $\mathbf{X}$ .)

### 3.1.2 Formulierung als Regressionsproblem

Wir widmen uns nun einer anderen Sichtweise auf die Hauptkomponentenanalyse, welche einen Zusammenhang zur linearen Regression herstellt und eine geometrische Interpretation ermöglicht. Hierbei möchte man einen  $k$ -dimensionalen Unterraum finden, der die Daten

bestmöglich approximiert wenn sie darauf projiziert werden. Mathematisch ausgedrückt minimieren wir also die Residuen der Projektion.

Sei dazu  $x_i$  die  $i$ -te Beobachtung, also die  $i$ -te Zeile von  $\mathbf{X}$  und  $\mathbf{V}_k = [v_1 \ \cdots \ v_k]$  eine  $p \times k$  orthonormale Matrix. Wie in Abschnitt 2.1.1 beschrieben wird durch den Operator  $\mathbf{V}_k \mathbf{V}_k^T$  jede Beobachtung orthogonal auf den durch  $V_1, \dots, V_k$  aufgespannten Unterraum projiziert. Eine bestmögliche  $\ell_2$ -Approximation der Daten ist gegeben wenn wir die Distanz zwischen jeder Beobachtung und seiner Projektion minimieren [ZHT06]:

$$\begin{aligned} \hat{\mathbf{V}}_k &= \arg \min_{\mathbf{V}_k} \sum_{i=1}^n \left\| x_i - \mathbf{V}_k \mathbf{V}_k^T x_i \right\|_2^2 \\ \mathbf{V}_k^T \mathbf{V}_k &= \mathbf{I}_{k \times k} \end{aligned} \quad (3.3)$$

Ein mathematisch rigoroser Beweis, dass die Lösung von (3.3) genau den ersten  $k$  Hauptachsen entspricht, befindet sich in [VMS16]. Wir möchten hier eine intuitive Erklärung für diese Äquivalenz geben. Wenn wir von der Varianz eines Datensatzes sprechen, reden wir von der Summe der Varianzen der einzelnen Variablen. Somit ist die Gesamtvarianz durch  $\frac{1}{n-1} \|\mathbf{X}\|_F^2$  gegeben. Mithilfe des verallgemeinerten Satzes von Pythagoras erhalten wir

$$\begin{aligned} \|\mathbf{X}\|_F^2 &= \sum_{i=1}^n \|x_i\|_2^2 \\ &= \sum_{i=1}^n \left\| x_i - \mathbf{V}_k \mathbf{V}_k^T x_i + \mathbf{V}_k \mathbf{V}_k^T x_i \right\|_2^2 \\ &= \sum_{i=1}^n \left\| x_i - \mathbf{V}_k \mathbf{V}_k^T x_i \right\|_2^2 + \left\| \mathbf{V}_k \mathbf{V}_k^T x_i \right\|_2^2 \\ &= \left\| \mathbf{X} - \mathbf{X} \mathbf{V}_k \mathbf{V}_k^T \right\|_F^2 + \left\| \mathbf{X} \mathbf{V}_k \mathbf{V}_k^T \right\|_F^2. \end{aligned}$$

Weiter sehen wir, dass

$$\begin{aligned} \left\| \mathbf{X} \mathbf{V}_k \mathbf{V}_k^T \right\|_F^2 &= \text{tr} \left( \mathbf{X} \mathbf{V}_k \mathbf{V}_k^T (\mathbf{X} \mathbf{V}_k \mathbf{V}_k^T)^T \right) \\ &= \text{tr} \left( \mathbf{V}_k^T \mathbf{X}^T \mathbf{X} \mathbf{V}_k \right) \\ &= \sum_{i=1}^k v_i^T (\mathbf{X}^T \mathbf{X}) v_i \end{aligned}$$

In (3.1) und (3.2) haben wir den Term  $v_i^T \Sigma v_i$ , der die Varianz der  $i$ -ten Hauptkomponente beschreibt, sequentiell maximiert. Drücken wir (3.1) und (3.2) in einem gemeinsamen Berechnungsproblem aus ergibt sich

$$\begin{aligned} \hat{\mathbf{V}}_k &= \arg \max_{\mathbf{V}_k} \sum_{i=1}^k v_i^T \Sigma v_i \\ \mathbf{V}_k^T \mathbf{V}_k &= \mathbf{I}_{k \times k} \end{aligned} \quad (3.4)$$

Kommen wir nun zurück auf die Zerlegung der Gesamtvarianz

$$\frac{1}{n-1} \|\mathbf{X}\|_F^2 = \frac{1}{n-1} \left\| \mathbf{X} - \mathbf{X} \mathbf{V}_k \mathbf{V}_k^T \right\|_F^2 + \sum_{i=1}^k v_i^T \Sigma v_i$$

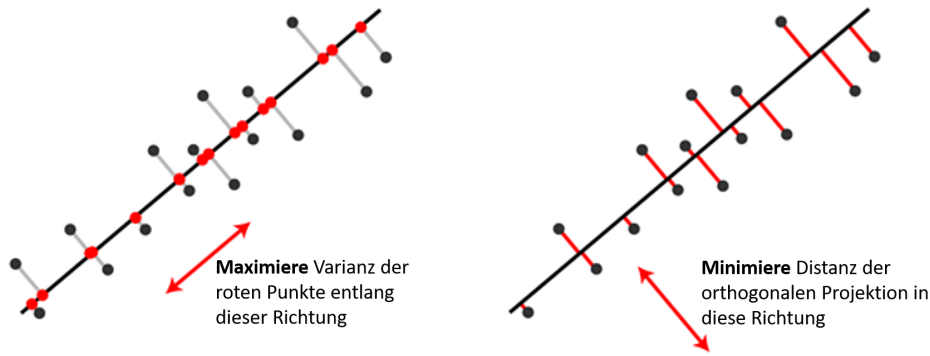


ABBILDUNG 3.2: Die Abbildung zeigt die Äquivalenz von Maximierung der Varianz und Minimierung der Residuen in zwei Dimensionen.

Mathematisch gesehen macht es keinen Unterschied, ob wir die Varianz der Hauptkomponenten  $\sum_{i=1}^k v_i^T \Sigma v_i$  maximieren oder die Residuen der Projektion  $\frac{1}{n-1} \|\mathbf{X} - \mathbf{X}\mathbf{V}_k\mathbf{V}_k^T\|_F^2$  minimieren. Diese Idee ist geometrisch in Abbildung 3.2 veranschaulicht. Hierbei versucht man im ersten Fall eine Hauptachse durch den Datensatz zu finden, so dass die Varianz entlang dieser maximiert werden. Anders versucht man im zweitem Fall eine Hauptachse zu finden, so dass die Summe der Distanzen zwischen Datenpunkt und seiner Projektion minimiert wird.

Da die Daten auf den niedrigdimensionaleren Raum linear transformiert werden gehört die Hauptkomponentenanalyse zu den linearen Dimensionsreduktionsverfahren. Anhand von (3.3) erkennt man zudem einen starken Zusammenhang zur linearen Regression, bei welcher ebenfalls die Summe der Residuenquadrate minimiert werden. Neben der unterschiedlichen Motivation der beiden Verfahren liegt die entscheidende Differenz in der Art der Projektion. Während bei linearer Regression die Projektion orthogonal bezüglich der unabhängigen Koordinatenachsen ist, werden die Daten bei der Hauptkomponentenanalyse orthogonal auf die Hauptachsen projiziert. Ausgehend von (3.3) werden wir im nächsten Kapitel die Variante der dünnbesetzten Hauptkomponentenanalyse beschreiben.

### 3.1.3 Weitere Formulierungen

Wir werden nun kurz auf zwei weiteren Formulierungen eingehen anhand welcher weitere Eigenschaften der Hauptkomponentenanalyse deutlich werden. Da  $\mathbf{Z}_k = \mathbf{X}\mathbf{V}_k$  die ersten  $k$  Hauptkomponenten beinhaltet, kann (3.3) auch geschrieben werden als

$$(\hat{\mathbf{Z}}_k, \hat{\mathbf{V}}_k) = \arg \min_{\mathbf{Z}_k, \mathbf{V}_k} \left\| \mathbf{X} - \mathbf{Z}_k \mathbf{V}_k^T \right\|_F^2 \quad (3.5)$$

$$\mathbf{V}_k^T \mathbf{V}_k = \mathbf{I}_{k \times k}.$$

Durch  $\mathbf{Z}_k \mathbf{V}_k^T$  ist also eine bestmögliche Rekonstruktion der Datenmatrix  $\mathbf{X}$  gegeben. In anderen Worten wird in (3.5) also der  $\ell_2$ -Rekonstruktionsfehler minimiert. Diese Formulierung werden wir als Grundlage für Verallgemeinerungen der Hauptkomponentenanalyse in Abschnitt 3.4 wählen.

Wir haben bereits gesehen, dass die abgeschnittene Singulärwertzerlegung  $\mathbf{X}_k = \mathbf{U}_k \mathbf{D}_k \mathbf{V}_k^T \in \mathbb{R}^{n \times k}$  die optimale Lösung für die Hauptkomponentenanalyse liefert. Diese ist aufgrund des Eckart-Young-Mirsky-Theorem, welches wir bereits in Abschnitt 2.1.3 kennengelernt haben,

Lösung des Problems

$$\begin{aligned} \hat{\mathbf{X}}_k &= \arg \min_{\mathbf{X}_k} \|\mathbf{X} - \mathbf{X}_k\|_F^2 \\ \text{rank}(\mathbf{X}_k) &\leq k. \end{aligned} \quad (3.6)$$

Somit ist  $\hat{\mathbf{X}}_k$  diejenige Matrix mit Rang  $k$ , die  $\mathbf{X}$  am Besten approximiert. Mithilfe dieser Formulierung können wir den Fehler, der durch die Dimensionsreduktion entsteht, explizit angeben:

$$\|\mathbf{X} - \hat{\mathbf{X}}_k\|_F^2 = \sum_{i=k+1}^p \sigma_i^2.$$

Infolgedessen können wir Bewertungskriterien definieren, welche es uns ermöglichen verschiedene Modelle miteinander zu vergleichen. Wir wollen uns nun mit der Frage beschäftigen, welche Hauptkomponenten eine geeignete Modellierung des Datensatzes darstellen.

## 3.2 Selektion der Hauptkomponenten

Optimale Hyperparameter für ein Modell zu finden ist selten einfach. Auch bei Dimensionsreduktionsverfahren kennen wir oft die intrinsische Dimension unserer Daten a priori nicht. Daher ist es schwer zu sagen, wie viele Hauptkomponenten benötigt werden, um die Daten passend zu modellieren. Es gilt einen Punkt der Balance zwischen Rekonstruktionsfehler und Modellkomplexität zu finden, welcher vom Anwendungsfall abhängen kann. Als Maß für die Modellkomplexität eignet sich aufgrund der natürlichen Ordnung in diesem Fall die Anzahl an Hauptkomponenten bzw. der Rang von  $\hat{\mathbf{X}}_k$ . Arbeiten wir auf rauschfreien Daten können wir  $k$  durch  $\text{rank}(\mathbf{X})$  schätzen. In der Regel ist unser Datensatz aber durch Rauschen gestört, weshalb  $\mathbf{X}$  vollen Rang hat.

Da für die Singulärwertzerlegung einer Matrix effiziente Algorithmen zur Verfügung stehen, berechnet man häufig zunächst alle  $k \leq \min\{n, p\}$  Hauptkomponenten. Die eigentliche Dimensionsreduktion findet dann durch Selektion statt. Bewertet werden die verschiedenen Modelle mithilfe des Rekonstruktionsfehlers bzw. der erklärten Varianz  $\sigma_i^2$  der einzelnen Hauptkomponenten. So können wir  $k$  zum Beispiel so wählen, dass der Rekonstruktionsfehler durch einen Parameter  $\tau$  beschränkt ist, d.h.

$$\hat{k} = \min_k \left\{ k: \sum_{i=k+1}^p \sigma_i^2 < \tau \right\} \quad \text{oder} \quad \hat{k} = \min_k \{k: \sigma_{k+1}^2 < \tau\}. \quad (3.7)$$

Das zweite Auswahlkriterium wird in Abbildung REF veranschaulicht. In der Praxis ist aber sehr schwer,  $\tau$  angemessen zu wählen, da die Singulärwerte von  $\mathbf{X}$  nicht invariant unter Skalierung sind. Daher werden die Singulärwerte meist normiert:

$$\hat{k} = \min_k \left\{ k: \frac{\sum_{i=k+1}^p \sigma_i^2}{\sum_{i=1}^k \sigma_i^2} < \tau \right\} \quad \text{oder} \quad \hat{k} = \min_k \left\{ k: \frac{\sigma_{k+1}^2}{\sum_{i=1}^k \sigma_i^2} < \tau \right\}. \quad (3.8)$$

Das erste Auswahlkriterium in (3.8) ist weit verbreitet. Man wählt genau so viele Hauptkomponenten aus, dass höchstens ein gewisser Anteil der Varianz verloren geht. Typische Werte für  $\tau$  liegen zwischen 10% und 20%.



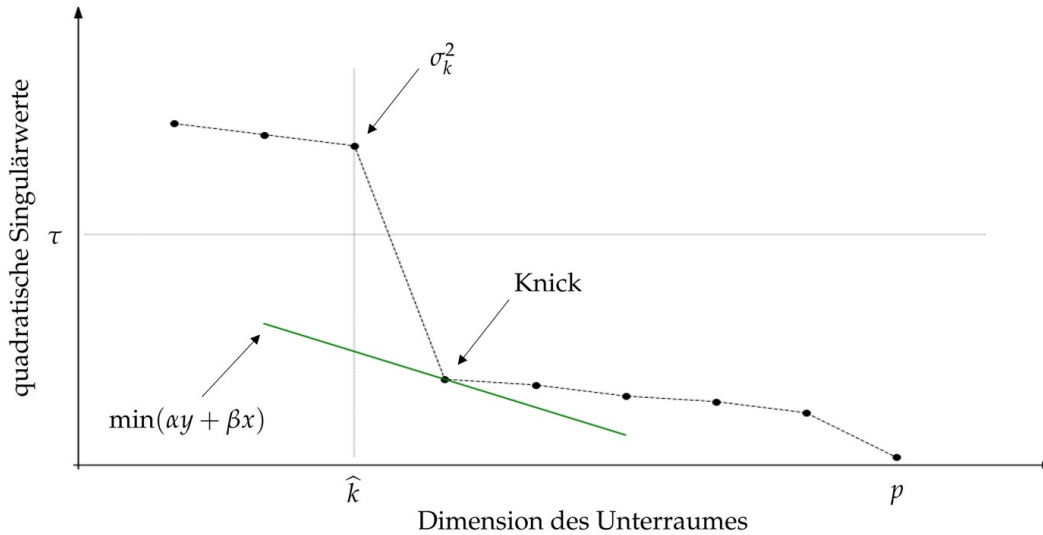


ABBILDUNG 3.3: Die Abbildung zeigt einen sog. *Scree Plot*, welche die quadratischen Singulärwerte bzw. die erklärte Varianz jeder einzelnen Hauptkomponente zeigt. Die Darstellung wird häufig genutzt, um zu entscheiden wie viele Hauptkomponenten für ein geeignetes Modell auszuwählen sind. (Abbildung basiert auf [VMS16])

In manchen Fällen gibt es eine klare Trennung in der Größe der Singulärwerte. Vidal et al. [VMS16] definieren ein Kriterium, welches nach einer solchen sucht:

$$\hat{k} = \arg \min_k \alpha \sigma_{k+1}^2 + \beta k \quad (3.9)$$

Mit  $\alpha, \beta > 0$  haben wir damit direkten Einfluss auf das Verhältnis zwischen Rekonstruktionsfehler und Modellkomplexität. In Abbildung REF haben wir die resultierende Gerade eingezeichnet, welche im wesentlichen nach einem Knick sucht. Allerdings möchten wir betonen, dass es nicht zwangsläufig einen solchen geben muss bzw. es auch mehrere geben kann.

In der Literatur existieren weitere Heuristiken auf die wir hier nicht näher eingehen werden. Dazu gehören beispielsweise ein *Bayes-Informationskriterium* (BIC) und ein *Akaike-Informationskriterium* (AIC) aus der Informationstheorie. Bis vor Kurzem gab es keinerlei theoretische Resultate für die optimale Modellwahl bei der Hauptkomponentenanalyse. Gavish und Donoho [GD14] haben erstmals ein optimales Kriterium im asymptotischem Fall angeben können, welche der Singulärwerte beibehalten werden sollten. Unter der Annahme, dass das Signal-Rausch-Verhältnis konstant bleibt, zeigen sie, dass der asymptotische mittlere quadratische Fehler

$$\text{AMSE} = \lim_{n \rightarrow \infty} \left\| \mathbf{X} - \hat{\mathbf{X}}_k \right\|_F^2$$

für  $n \times n$ -Matrizen minimal wird, falls alle Singulärwerte kleiner als  $\frac{4}{\sqrt{3}} \sqrt{n} \sigma \approx 2.309 \sigma$  auf Null gesetzt werden. Wenn die Stärke des Rauschens  $\sigma$  vorher nicht bekannt ist, ändert sich die Schranke zu  $2.858 \sigma_{med}$ , wobei  $\sigma_{med}$  der Stichprobenmedian der Singulärwerte ist. Für nicht quadratische  $m \times n$ -Matrizen ändern sich die Schwellenwerte abhängig von  $m, n$ . Auch wenn damit nur ein asymptotisches Resultat zur Verfügung steht, scheint diese Schranke in der Praxis hilfreich zu sein.



### 3.3 Grenzen der Anwendbarkeit

Obwohl die Hauptkomponentenanalyse in vielen Situationen helfen kann, Datensätze zu veranschaulichen und zu strukturieren, gibt es keine Garantie für sinnvolle Ergebnisse. Im Folgenden werden wir Szenarien beschreiben, bei denen unerwünschte Effekte bei der Verwendung dieses Verfahrens auftreten. Daher gilt es den Datensatz vorerst hinsichtlich folgender Gesichtspunkte zu untersuchen:

- Lineare Beziehung zwischen Variablen
- Vollständigkeit des Datensatzes
- Ausreißer in den Daten
- Anzahl an Beobachtungen in Relation zu Anzahl an Variablen

Bei der Hauptkomponentenanalyse können lineare Korrelationen zwischen Variablen sehr gut eingefangen werden. Andere Beziehungen, die durchaus in der Praxis vorkommen, werden dabei nicht berücksichtigt bzw. als linear angenommen. Vidal et al. [VMS16] zeigen am Beispiel von Porträt-Fotos auf, dass nichtlineare Strukturen in der Tat verloren gehen. In manchen Fällen können wir mithilfe geeigneter Koordinatentransformationen dieses Hindernis umgehen. Bei der Anwendung des sog. *Kernel Tricks* transformiert man die Daten in einen höherdimensionalen Raum, in welchem sie besser linear separierbar sind. Bei *Kernel PCA* wendet man anschließend die Hauptkomponentenanalyse in diesem Raum an bevor man die Daten wieder zurück transformiert. Andere Erweiterungen, die unter der Kategorie *manifold learning* zusammengefasst werden können, versuchen die lokale Geometrie der Mannigfaltigkeit direkt zu approximieren. So kann man ohne die Nutzung eines höherdimensionalen Raumes unmittelbar die niedrigdimensionale Struktur erhalten. Verfahren, die diesen Ansatz verfolgen, sind zum Beispiel die *multidimensionale Skalierung* oder *ISOMAP*. [LV07]

Falls wir einen unvollständigen Datensatz vorliegen haben, bei welchem manche Werte korrupt oder nicht vorhanden sind, können wir die klassische Hauptkomponentenanalyse nicht anwenden. Nehmen wir an, dass dies nur einen Bruchteil der Daten betrifft, können mit hoher Wahrscheinlichkeit noch immer exakte Ergebnisse erzielt werden [Can+11]. NP hardness Gillis Low-Rank Matrix Approximation with Weights or Missing Data Is NP-Hard. Des Weiteren ist das Verfahren sehr sensitiv gegenüber Ausreißern. Besonders wenn nur wenige Beobachtungen zur Verfügung stehen, können Ausreißer die Resultate drastisch beeinflussen. Um eine Verfälschung der Ergebnisse zu vermeiden, entfernt man diese meist vor der Anwendung. Allerdings ist es nicht immer einfach Ausreißer zu identifizieren. Daher wird in [Kri+08] vorgeschlagen, die Datenpunkte basierend auf deren geschätzten Relevanz vorab unterschiedlich zu gewichten. Die Behandlung dieser Art Probleme wird mit dem Begriff *Robust PCA* verbunden.

Viele gegenwärtige Datensätze besitzen eine vergleichsweise hohe Anzahl an Variablen im Vergleich zur Anzahl an Beobachtungen. Typische Anwendungsgebiete sind die Bildbearbeitung, Signal- oder Genexpressionsanalyse, in welchen viele Variablen in Form von Pixeln, Frequenzen oder Genen vorkommen. Auch wir werden uns in Kapitel 6 mit einem Datensatz dieser Art beschäftigen für welchen  $p \approx 50000$  und  $n = 33$  ist. In Theorem ?? werden wir sehen, dass es in einer solchen Situation zur *Inkonsistenz* der Hauptkomponentenanalyse kommen kann. Deswegen muss in diesen Fällen genau überprüft werden, ob den Ergebnissen getraut werden kann.

Das wohl wichtigste Hindernis im Zuge dieser Arbeit sind sicherlich die durch die Transformation entstehenden Interpretationsschwierigkeiten. Da jede Hauptkomponente durch

Verlustfunktion	Regularisierung	Nebenbedingung
quadratisch (reelle Daten)	$\ell_2$ -Norm (kleine Koeffizienten)	orthogonal (orthogonale Hauptachsen)
absolut (robust gegenüber Ausreißern)	$\ell_1$ -Norm (dünnbesetzte Koeffizienten)	nichtnegativ (nichtnegative Koeffizienten)
logistisch (Binärdaten)	Ableitung Strafterm (glatte Koeffizienten)	

TABELLE 3.1: Zusammenfassung einiger Erweiterungen der Hauptkomponentenanalyse in einem konsistenten Schema.

eine Linearkombination aller Ausgangsvariablen entsteht, ist es in hochdimensionalen Fällen fast unmöglich diesen eine Bedeutung im Kontext zuzuweisen. Dieser Interpretationsverlust ist Ausgangspunkt der Idee der dünnbesetzten Hauptkomponentenanalyse (kurz: *Sparse PCA*). Diesem Verfahren ist das folgende Kapitel 4 gewidmet.

### 3.4 Erweiterungen der Hauptkomponentenanalyse

Im vorangegangenen Abschnitt haben wir bereits auf Erweiterungen der Hauptkomponentenanalyse hingewiesen, welche Grenzen der klassischen Variante überkommen. Viele dieser können wir mithilfe eines allgemeinen Schemas formulieren.

$$\arg \min_{\mathbf{Z}, \mathbf{V}} \underbrace{L(\mathbf{Z}, \mathbf{V})}_{\text{Verlustfunktion}} + \underbrace{\lambda_z f_z(\mathbf{Z}) + \lambda_v f_v(\mathbf{V})}_{\text{Regularisierung}} \quad (3.10)$$

unter der Nebenbedingung, dass  $\underbrace{\mathbf{Z} \in \Omega_z, \mathbf{V} \in \Omega_v}_{\text{Nebenbedingung}}$

In Tabelle 3.1 sind einige Ideen, die in der Literatur betrachtet werden, festgehalten. Dabei können Verlustfunktion, Regularisierung und Nebenbedingungen beliebig kombiniert werden, um ein für den Anwendungsfall geeignetes Modell zu erstellen. (Möchte man an der Stelle sehen, wie die einzelnen Verlustfunktionen aussehen?)

### 3.5 Theoretische Aussagen

Zu Abschluss dieses Kapitels werden wir einige interessante Eigenschaften sowie Theoreme präsentieren. Der Fokus wird vor allem auf der Inkonsistenz der Hauptkomponentenanalyse in hochdimensionalen Fällen sein.

Eine wichtige Eigenschaft ist die Unkorreliertheit der Hauptkomponenten. Dies lässt sich leicht anhand der Stichprobenkovarianzmatrix der Hauptkomponenten  $\frac{1}{n-1} \mathbf{Z}^T \mathbf{Z} = \frac{1}{n-1} (\mathbf{XV})^T \mathbf{XV} = \mathbf{V}^T \mathbf{\Sigma} \mathbf{V}$  sehen, welche aufgrund von  $\mathbf{V}^T \mathbf{V} = \mathbf{I}_{k \times k}$  eine Diagonalmatrix ist. Somit können wir von einer Hauptkomponente und deren erklärte Varianz reden ohne uns dabei auf andere beziehen zu müssen.

Unter der Annahme, dass wir uniqueness of PCA?

Falls mehr Variablen ausgewählt werden sollen, no recomputation necessary.

Für unseren Anwendungsfall in Kapitel 6 verfügen wir nur über wenig Beobachtungen im Vergleich zu der Anzahl an Variablen im Datensatz. Es gibt eine Reihe theoretischer Aussagen, welche eine Inkonsistenz im asymptotischem Fall zeigen. Asymptotische Studien der Hauptkomponentenanalyse fallen in verschiedene Kategorien abhängig vom Verhältnis zwischen  $n$  und  $p$ .

- In der klassischen Asymptotik untersucht man die Effekte für  $n \rightarrow \infty$  mit fixierter Dimension  $p$ .
- In der Theorie der Zufallsmatrizen erhöhen wir sowohl  $n$  als auch  $p$  mit der Voraussetzung, dass das Verhältnis  $\frac{p_n}{n} \rightarrow c$  konstant ist.
- In einem *high dimension low sample size setting* (HDLSS) betrachten wir die Asymptotik für  $p \rightarrow \infty$  mit fixierter Anzahl an Beobachtungen  $n$ .

Zunächst gilt es den Begriff der *Inkonsistenz* in Bezug auf die Hauptkomponentenanalyse genauer zu verstehen. In dieser Arbeit haben wir eine Stichprobenversion der Hauptkomponentenanalyse eingeführt. Indem man die Stichprobenkovarianzmatrix  $\Sigma$  durch die Kovarianzmatrix in (3.1) ersetzt, erhält man eine Populationsversion der Hauptkomponentenanalyse. Unter der Annahme, dass die gegebenen Daten ein endliche, zufällige Stichprobe einer (meist unbekannten) Verteilung sind, stellt sich die Frage in welcher Relation die Ergebnisse stehen. Genauer gesagt interessiert man sich dafür, ob die Hauptachsen der Stichproben- und der Populationsversion identisch sind. Um die Hauptachsen miteinander zu vergleichen beziehen wir uns auf den Winkel zwischen Beiden.

**Definition 3.1** (Konsistenz [SSM13]). Die Stichprobenhauptachse  $\bar{v}_i$  ist mit der Populationshauptachse  $v_i$

- *konsistent*, falls  $\text{angle}(\bar{v}_i, v_i) \rightarrow 0$
- *marginal inkonsistent*, falls  $\text{angle}(\bar{v}_i, v_i) \rightarrow c \in (0, \frac{\pi}{2})$
- *stark inkonsistent*, falls  $\text{angle}(\bar{v}_i, v_i) \rightarrow \frac{\pi}{2}$

für  $n \rightarrow \infty$  bzw.  $p \rightarrow \infty$ .

Abhängig von  $n$ ,  $p$  und der Stärke des Rauschens im Modell ergeben sich unterschiedliche Ergebnisse für die Konsistenz der Hauptkomponentenanalyse. Allgemein ergibt sich die Tendenz, dass eine hohe Anzahl an Beobachtungen die Konsistenz der Hauptachsen fördert, während eine erhöhte Dimension die Inkonsistenz unterstützt. Nicht überraschend kann man zeigen, dass die Eigenvektoren der Stichprobenkovarianzmatrix  $\bar{v}_i$  konsistente Schätzer für die Eigenvektoren der Kovarianzmatrix  $v_i$  sind, falls wir die Dimension fixieren und die Anzahl an Beobachtungen erhöhen  $n \rightarrow \infty$  [And03]. Für unser Szenario jedoch interessanter sind hochdimensionale Fälle. Zum Abschluss dieses Kapitels präsentieren wir daher zwei Inkonsistenz-Theoreme.

Für den Fall eines asymptotisch konstantem Verhältnis zwischen  $n$  und  $p$  betrachten wir ein *single component model* [JY09]

$$x_i = l_i v_1 + \sigma r_i \quad i = 1, \dots, n, \quad (3.11)$$

in welchem  $v_1 \in \mathbb{R}^p$  der zu schätzende Eigenvektor ist. Dabei sind  $l_i \sim N(0, 1)$  unabhängig gleichverteilte Gaußsche Zufallseffekte und  $r_i \sim N_p(0, \mathbf{I})$  unabhängige Rauschvektoren. Wegen der Rauscheffekte hat die Stichprobenkovarianzmatrix  $\Sigma$  fast sicher  $\min\{n, p\}$  von null verschiedene Eigenvektoren. Sei  $\bar{v}_1$  der größte Stichprobeneigenvektor von  $\Sigma$ . Mit dem folgendem Theorem haben wir eine notwendige und hinreichende Bedingung für die Konsistenz der ersten Stichprobenhauptachse.

**Theorem 3.2** (Inkonsistenz der Hauptkomponentenanalyse bei konstantem asymptotischem Verhältnis von  $n$  und  $p$  [JY09]). Sei  $\sigma$  die Stärke des Rauschens im Modell (3.11) und  $\omega = \lim_{n \rightarrow \infty} \frac{\|v_1\|^2}{\sigma^2} > 0$  das limitierende Signal-Rausch-Verhältnis. Weiterhin sei

$$R(\bar{v}_1, v_1) = \left\langle \frac{\bar{v}_1}{\|\bar{v}_1\|}, \frac{v_1}{\|v_1\|} \right\rangle = \cos(\text{angle}(\bar{v}_1, v_1)).$$

Unter der Annahme, dass  $\lim_{n \rightarrow \infty} \frac{p_n}{n} = c$  gilt

$$\mathbb{P} \left[ \lim_{n \rightarrow \infty} R^2(\bar{v}_1, v_1) = R_\infty^2(\omega, c) \right] = 1$$

wobei  $R_\infty^2(\omega, c) = \frac{(\omega^2 - c)_+}{\omega^2 + c\omega}$ .

Wir sehen, dass  $R_\infty^2(\omega, c) < 1$  genau dann, wenn  $c > 0$  und somit ist  $\bar{v}_1$  ein konsistenter Schätzer für  $v_1$  genau dann, wenn  $c = 0$ . Für  $c > 0$  ist die Hauptkomponentenanalyse daher inkonsistent.

Ein sehr ähnliches Resultat hält auch für ein sog. *spiked covariance model*, in welcher mehrere Komponenten betrachtet werden [Pau07]. Des Weiteren zeigen Jung und Marron eine starke Inkonsistenz der Stichprobenhauptachsen in einer HDLSS-Situation unter geeigneten Modellannahmen. Dabei kann Konsistenz nur dann gewährleistet werden, wenn die führenden Eigenwerte klar größer sind als die Restlichen und weitere Eigenschaften gewährleistet sind. Für Details verweisen wir auf [JM09].

## Kapitel 4

# Dünnbesetzte Hauptkomponentenanalyse

Ein wesentlicher Nachteil der Hauptkomponentenanalyse besteht darin, dass sich die neuen Variablen aus einer Linearkombination *aller* bestehenden Variablen zusammensetzt. Dies erschwert besonders für hochdimensionale Daten eine Interpretation der Hauptachsen. Während zuvor jede Variable eine Bedeutung hatte, sind wir nach der Transformation meist nicht in der Lage den Hauptachsen eine Bedeutung im Kontext zuzuweisen. Um zu verstehen, was die Hauptachsen im Modell repräsentieren kann es besonders hilfreich sein, wenn diese *dünnbesetzt* sind, sich also nur aus wenigen Variablen zusammensetzen. Treffen wir irgendwelche Annahmen? Des Weiteren ist nicht jede Variable relevant zur Strukturerkennung. impose extra constraints, which sacrifices some variance in order to improve interpretability. Interpretation ist oberstes Ziel!!!

shape/image analysis, ecological studys und neuroscience data application.

Zu Anfang dieses Kapitels werden wir eine naheliegende mathematische Formulierung des Problems beschreiben. Leider wird sich diese als NP-vollständig herausstellen, weshalb wir in Abschnitt 4.2 verschiedene Wege aufzeigen, dass Problem zu relaxieren. In 4.3 möchten wir uns mit einem dieser Ansätze intensiv beschäftigen, welcher den Ausgangspunkt für den weiteren Verlauf dieser Arbeit darstellt. Wir haben uns dazu entschieden, den von Zou, Hastie und Tibshirani in [ZHT06] eingeführten Ansatz für diesen Zweck zu verfolgen. Dieser gilt sicherlich zu der am meisten verbreiteten Variante der dünnbesetzten Hauptkomponentenanalyse. Kürzlich erschienenenes Paper, Neuerungen. Der Rest dieses Kapitels ist den Details dieses Ansatzes gewidmet.

### 4.1 Problemformulierung

Wir möchten nun Hauptachsen eines gegebenen Datensatzes identifizieren mit der Zusatzbedingung, dass diese dünnbesetzt sind. Die wohl einfachste Vorgehensweise ist, zuerst die gewöhnliche Hauptkomponentenanalyse durchzuführen und anschließend ein Schwellwertmethode auf die Hauptachsen anzuwenden. Hierbei vernachlässigt man alle Koeffizienten, die kleiner als ein bestimmter Schwellenwert sind, indem man sie auf 0 setzt. Eine solche Prozedur kann aber in vielen Fällen irreführend sein, unter welcher die Qualität der Ergebnisse leidet [CJ95]. Die Wichtigkeit einer Variable in den Hauptachsen wird nicht allein durch den Koeffizient bestimmt. Zu berücksichtigen sind unter anderem sowohl die Standardabweichung als auch die Korrelationen mit anderen Variablen. Bei einer Schwellwertmethode werden diese Faktoren nicht beachtet, weshalb den Ergebnissen im Allgemeinen nicht vertraut werden darf.

Hier Regression on ordinary PCA's mit Zou et al?

Anstelle eines zweischrittigen Ansatzes kann die Dünnbesetzung direkt in die Problemformulierung mit eingebaut werden. Gegeben sei dazu wieder eine Datenmatrix  $\mathbf{X} \in \mathbb{R}^{n \times p}$ , wobei  $n$  die Anzahl an Beobachtungen und  $p$  die Anzahl an Variablen ist. Des Weiteren gehen wir davon aus, dass die Matrix  $\mathbf{X}$  zuvor spaltenweise zentriert wurde. Dann kann die dünnbesetzte Hauptkomponentenanalyse als sukzessives Maximierungsproblem formuliert werden:

$$\begin{aligned} v_k &= \arg \max_{\|v\|_2=1} v^T \Sigma v \\ \text{unter der Nebendingung, dass für } k \geq 2 \quad &v_k^T v_l = 0 \quad \forall 1 \leq l < k \\ &\text{und } \|v_k\|_0 \leq t \end{aligned} \quad (4.1)$$

wobei  $\Sigma = \frac{\mathbf{X}^T \mathbf{X}}{n-1}$  die Stichprobenkovarianzmatrix ist. Der einzige Unterschied zur klassischen Hauptkomponentenanalyse, wie wir sie in (3.1) beschrieben haben, besteht in der Einführung der  $\ell_0$ -Norm. Somit beschränken wir uns auf die Suche von Hauptachsen, welche höchstens  $t$  von Null verschiedene Einträge haben. Wählen wir  $t = p$  reduziert sich das Problem auf (3.1). Während (4.1) eine sehr schöne und einfache mathematische Formulierung ist, wurde gezeigt, dass dieses Problem NP-vollständig ist [FR13]. Zur Berechnung dünnbesetzter Hauptachsen sind wir also angehalten eine geeignete Relaxation zu finden.

## 4.2 Relaxation

Es existiert eine Vielfalt an Ansätzen, um das Problem zu relaxieren. Wir wollen zunächst einen kleinen Überblick über die unterschiedlichen Ideen geben und uns anschließend mit einer genauer beschäftigen. Eine selektive Übersicht der verschiedenen Ansätze haben wir hier erstellt.

### SCoTLASS

Inspiziert von der Lasso Regression [Tib96] schlugen Jolliffe et al. [JTU03] vor, die  $\ell_1$ -Norm anstelle der  $\ell_0$ -Norm als Strafterm zu verwenden. Wie wir bereits in Abschnitt 2.3.4 gesehen haben, kann die  $\ell_1$ -Norm genutzt werden, um dünnbesetzte Vektoren zu erhalten. Somit liegt es nahe das Problem wie folgt zu formulieren.

$$\begin{aligned} v_k &= \arg \max_{\|v\|_2=1} v^T \Sigma v \\ \text{unter der Nebendingung, dass für } k \geq 2 \quad &v_k^T v_l = 0 \quad \forall 1 \leq l < k \\ &\text{und } \|v_k\|_1 \leq t \end{aligned} \quad (4.2)$$

Wie in (4.1) hat man mit der Wahl der Parameters  $t$  Einfluss auf die Dünnbesetzung der Hauptachsen. Aufgrund der hohen Berechnungskosten ist SCoTLASS allerdings für hochdimensionale Daten ungeeignet. Diese sind vor allem darauf zurückzuführen, dass (4.2) kein konvexes Optimierungsproblem ist. Des Weiteren ergeben sich Schwierigkeiten bei der Wahl des Hyperparameters  $t$ . Auch wenn eine passende Wahl eine gewünschte Dünnbesetzung hervorruft, gibt es kaum Orientierungshilfen. Zusätzlich hat SCoTLASS dasselbe grundlegende Problem wie das Lasso. Die Anzahl von null verschiedener Einträge ist durch die Anzahl Beobachtungen im Datensatz limitiert, welches die Brauchbarkeit des Modells deutlich einschränkt. Zusammen mit den hohen Berechnungskosten ist dieser Ansatz in der

Praxis daher meist impraktikabel.

### Semidefinite Programmierung

Konvexe Relaxation ist eine Standard-Technik, um mit schwierigen nichtkonvexen Problemen umzugehen. d'Aspremont et al. [dAs+07] entwickeln einen Ansatz, welcher sich als semidefinites Programmierungsproblem ausdrücken lässt. Zunächst werden wir (4.1) dafür mit Matrizen reformulieren.

Sei  $\mathbf{V} = v_k v_k^\top$ . Dann übersetzen sich die Nebenbedingungen

$$\begin{aligned} \arg \max_{\mathbf{P}} &= \text{tr}(\mathbf{\Sigma P}) \\ \text{tr}(\mathbf{P}) &= 1, \quad \|\mathbf{P}\|_0 \leq k^2, \quad \mathbf{P} \geq 0, \quad \text{rank}(\mathbf{P}) = 1 \end{aligned} \quad (4.3)$$

Diese Formulierung ist noch immer nichtkonvex aufgrund der Rang-Bedingung und der  $\ell_0$ -Strafterm. Per Definition ist  $\mathbf{P}$  symmetrisch und  $\mathbf{P}^2 = \mathbf{P}$ . Somit ist

$$\|\mathbf{P}\|_F^2 = \text{tr}(\mathbf{P}^\top \mathbf{P}) = \text{tr}(\mathbf{P}) = 1$$

und mit der Cauchy-Schwarz-Ungleichung folgt

$$\mathbf{1}_p^\top |\mathbf{P}| \mathbf{1}_p \leq \sqrt{\|\mathbf{P}\|_0 \|\mathbf{P}\|_F^2} \leq k$$

Ersetzen wir die  $\ell_0$ -Strafterm und lassen die Rang-Bedingung fallen erhalten wir die DSPCA-Formulierung

$$\begin{aligned} \arg \max_{\mathbf{P}} &= \text{tr}(\mathbf{\Sigma P}) \\ \text{tr}(\mathbf{P}) &= 1, \quad \mathbf{1}_p^\top |\mathbf{P}| \mathbf{1}_p \leq k, \quad \mathbf{P} \geq 0 \end{aligned} \quad (4.4)$$

Dies stellt ein semidefinites Programmierungsproblem dar, bei welcher die zu optimieren- den Variablen symmetrische Matrizen sind unter der Nebenbedingung, dass sie positiv semidefinit sind. Für kleine Probleme kann (4.4) effizient durch *Innere-Punkte-Verfahren* (Eng- lish: *interior-point methods*) gelöst werden. SDPT3 [TTT99].

In (4.4) wird allerdings  $\mathbf{P}$  berechnet und nicht die eigentliche Hauptachse. Hierfür kürzen d'Aspremont et al. die Matrix  $\mathbf{P}$  und behalten nur den größten Eigenvektor  $v_k$ . Anschließend erhält man weitere Hauptachsen durch Matrix Deflation, indem wir  $\mathbf{\Sigma}$  durch

$$\mathbf{\Sigma} - (v_k^\top \mathbf{\Sigma} v_k) v_k v_k^\top$$

ersetzen. Für größere Probleme wird eine Methode von Nesterov benutzt, um eine Laufzeit von  $\mathcal{O}(\frac{p^4 \sqrt{\log p}}{\epsilon})$  zu erreichen.

### Iterative Schwellenwert-Methode



Basierend auf der Formulierung (3.6) der Hauptkomponentenanalyse als beste Rang  $k$  Approximation an die Datenmatrix  $\mathbf{X}$  haben Shen und Huang [SH08] das folgende Optimierungsproblem formuliert

$$(u_1, v_1) = \arg \min_{u,v} \left\| \mathbf{X} - uv^\top \right\|_F^2 + \lambda \|v\|_1 \quad (4.5)$$

$$\|u\|_2 = 1$$

Somit erhält man mit  $\frac{v_1}{\|v_1\|}$  die erste dünnbesetzte Hauptachse. Auch hier werden die restlichen Hauptachsen sequentiell berechnet durch Ersetzen der Datenmatrix  $\mathbf{X}_{(k+1)} = \mathbf{X} - \sum_{i=1}^k u_i v_i^\top$ . Jede Iteration kann durch ein alternierendes Minimierungsverfahren gelöst werden. Fixiert man  $v$ , so ist das optimale  $u$  gegeben durch  $u = \frac{\mathbf{X}v}{\|\mathbf{X}v\|}$ . Andererseits reduziert sich (4.5) für festes  $u$  auf

$$\arg \min_v -2\text{tr} \left( \mathbf{X}^\top uv^\top \right) + \|v\|^2 + \lambda \|v\|_1.$$

Eine explizite Lösung ist durch den soft-thresholding Operator gegeben

$$v = \text{soft}_{\frac{\lambda}{2}}(\mathbf{X}^\top u)$$

welcher in Abschnitt 2.3 eingeführt worden ist.

Diese Methode ist sehr ähnlich zu der von Zou et al. [ZHT06], mit welcher wir uns im folgenden Abschnitt beschäftigen werden. Der große Unterschied besteht darin, dass die Hauptachsen dort nicht sequentiell, sondern gleichzeitig berechnet werden. Witten et al. haben in [WTH09] ebenfalls eine Methode entwickelt, die unter diese Kategorie fällt.

### Weitere Relaxationsideen

Es gibt noch eine Reihe weiterer Ideen, die in der Literatur betrachtet wurden. Ein interessierter Leser sei auf die folgenden Ansätze verwiesen.

- eine verallgemeinerte Potenzmethode [Jou+10]
- ein alternierendes Maximierungs-Netzwerk [Ric12]
- Vorwärts und Rückwärts-Greedy-Suche mittels Branch-and-Bound-Verfahren [MWA06]
- eine Bayes-Formulierung [GD09]

## 4.3 Konstruktion

Wir werden uns nun mit dem von Zou, Hastie und Tibshirani in [ZHT06] eingeführten Ansatz ausführlich beschäftigen. Zou und Hastie führten zuvor in [ZH05] das sog. *elastic net* ein, welches den Grundstein für die mathematische Formulierung legen wird.

Wie bereits in (3.3) beschrieben kann die Hauptkomponentenanalyse als regressionsartiges Problem betrachtet werden. Das folgende Theorem erweitert die bisherige Formulierung, indem nun nicht ausschließlich orthogonale Projektionen erlaubt werden. Im Folgenden bezeichnet  $k$  die Anzahl an Hauptkomponenten, die wir extrahieren möchten und  $x_i$  die  $i$ -te Zeile von  $\mathbf{X}$ . Wir bezeichnen mit  $\mathbf{B}$  im Folgenden die dünnbesetzten Hauptachsen, um sie von den klassischen Hauptachsen  $\mathbf{V}$  zu unterscheiden.



**Theorem 4.1.** Sei  $\mathbf{A}_{p \times k} = [\alpha_1, \dots, \alpha_k]$  und  $\mathbf{B}_{p \times k} = [\beta_1, \dots, \beta_k]$ . Für ein  $\lambda_2 > 0$  sei

$$(\hat{\mathbf{A}}, \hat{\mathbf{B}}) = \arg \min_{\mathbf{A}, \mathbf{B}} \sum_{i=1}^n \left\| x_i - \mathbf{A} \mathbf{B}^T x_i \right\|^2 + \lambda_2 \sum_{j=1}^k \|\beta_j\|^2$$

wobei  $\mathbf{A}^T \mathbf{A} = I_{k \times k}$ .

Dann ist  $\hat{\beta}_j \propto v_j$  für  $j = 1, 2, \dots, k$ .

Fordern wir  $\mathbf{A} = \mathbf{B}$  reduziert sich die Verlustfunktion  $\sum_{i=1}^n \left\| x_i - \mathbf{A} \mathbf{B}^T x_i \right\|^2$  auf die klassische Hauptkomponentenanalyse (3.3). Theorem 4.1 zeigt, dass wir die Bedingung  $\mathbf{A} = \mathbf{B}$  unter Einführung eines Ridge-Strafterms vernachlässigen können. Mithilfe dieser Verallgemeinerung können wir die Hauptkomponentenanalyse flexibel modifizieren.

Um dünnbesetzte Hauptachsen zu erhalten können wir einen  $\ell_1$ -Strafterm in die Zielfunktion einbetten. Das ein solcher Strafterm eine gewünschte Dünnbesetzung hervorruft, haben wir in Abschnitt 2.3.4 beobachten können. Wir definieren daher das *Sparse PCA Kriterium* mit den Hyperparameter  $\lambda_{1,j}$  und  $\lambda_2$

$$(\hat{\mathbf{A}}, \hat{\mathbf{B}}) = \arg \min_{\mathbf{A}, \mathbf{B}} \sum_{i=1}^n \left\| x_i - \mathbf{A} \mathbf{B}^T x_i \right\|_2^2 + \lambda_2 \sum_{j=1}^k \|\beta_j\|_2^2 + \sum_{j=1}^k \lambda_{1,j} \|\beta_j\|_1 \quad (4.6)$$

unter der Nebenbedingung, dass  $\mathbf{A}^T \mathbf{A} = I_{k \times k}$

Die normierten Spalten von  $\mathbf{B}$  nennen wir dann die *dünnbesetzten Hauptachsen*. Um die Dünnbesetzung für jede Hauptachse unterschiedlich wählen zu können, erlauben wir unterschiedliche Bestrafungen  $\lambda_{1,j}$ . Dagegen erlauben wir für die Ridge-Bestrafung  $\lambda_2$ , die im Wesentlichen für die Reduktion von (4.6) auf (3.3) benötigt wird falls  $\lambda_{1,j} = 0$ , keine differenzierte Behandlung. Allerdings hat die  $\ell_2$ -Bestrafung noch einen weiteren Vorteil, welcher in der Praxis relevant ist. Es bewältigt das Lasso-Defizit, so dass auch mehr als  $n$  Variablen im Fall  $p > n$  ausgewählt werden können.

Wir möchten darauf hinweisen, dass im Gegensatz zu manch anderen Varianten der dünnbesetzten Hauptkomponentenanalyse (4.6) eine zeitgleiche anstatt einer sequentiellen Berechnung der Hauptachsen ermöglicht. Dies wird im folgendem Abschnitt von entscheidender Bedeutung sein.

## 4.4 Anpassung der Transformation, Residuen und Varianzen

Bei der Verwendung der dünnbesetzten Hauptkomponentenanalyse übertragen sich viele der Eigenschaften der klassischen Variante nicht. [Cam+20]. Daher gilt es folgende Punkte zu berücksichtigen.

### Korrelation der Hauptkomponenten

Bei einer klassischen Hauptkomponentenanalyse sind die Hauptkomponenten aufgrund der orthogonalen Hauptachsen unkorreliert. Letztere Eigenschaft fordern wir bei der dünnbesetzten Variante in (4.6) nicht, so dass durchaus starke Korrelationen zwischen den Hauptkomponenten auftreten können. Während dies eine flexiblere Modellierung ermöglicht, wird

es dadurch schwieriger die Ergebnisse geeignet zu visualisieren. Besonders bei der Verwendung von Streudiagrammen, welche genutzt werden, um den Beitrag der Ausgangsvariablen zu den Hauptachsen zu visualisieren, kann dies zu Problemen führen. Hierbei unterstellt man die Orthogonalität der Hauptachsen, was zu Verzerrungen der Distanzen im Bild führen kann [GML03]. Des Weiteren kann die Berechnung der erfassten Varianz des Datensatzes, welches häufig als Maß für die Qualität eines Modells genutzt wird, nicht analog zur klassischen Variante durchgeführt werden.

### Varianzverlust der Hauptkomponenten

Der Erfolg der Hauptkomponentenanalyse beruht vor allem darauf, dass die Hauptkomponenten optimal bezüglich erklärter Varianz sind. Oft kann ein Großteil an Information eines Datensatzes durch eine geringe Anzahl an Hauptkomponenten beschrieben werden, welches eine Visualisierung und Interpretation hochdimensionaler Daten ermöglicht. Bei der dünnbesetzten Hauptkomponentenanalyse opfern wir einen Teil der erklärten Varianz für simplere, einfacher zu interpretierende Hauptachsen. Um einen genauso großen Teil an Information des Datensatzes zu erklären, benötigen wir daher eine größere Anzahl an Hauptkomponenten in unserem Modell. Somit können wir die Dimension des Datensatzes unter Umständen nicht all zu stark reduzieren.

In einem erst vor Kurzem erschienen wissenschaftlichen Artikel zeigen Camacho et al. [Cam+20], dass viele der Varianten der dünnbesetzten Hauptkomponentenanalyse bezüglich der genannten Punkte Probleme aufweisen. Insbesondere wurde die Berechnung der Hauptkomponenten, Residuen und der erklärten Varianz bislang falsch durchgeführt. Wir möchten an dieser Stelle die Unterschiede detailliert erklären.

Typischerweise wurden bislang die Hauptkomponenten  $\mathbf{Z}$  wie bei der klassischen Hauptkomponentenanalyse berechnet, indem man  $\mathbf{Z} = \mathbf{X}\mathbf{B}$  setzt, wobei  $\mathbf{B}$  die Matrix der dünnbesetzten Hauptachsen ist [ZHT06]. Allerdings vernachlässigt man in diesem Fall, dass die Hauptachsen nicht orthogonal zueinander sind. Dies wird deutlich, wenn wir die Hauptkomponentenanalyse wie in (3.5) als eine bestmögliche Rekonstruktion der Datenmatrix  $\mathbf{X}$  auffassen.

$$\mathbf{X} = \mathbf{Z}\mathbf{B}^T + \mathbf{E}, \quad (4.7)$$

wobei  $\mathbf{E}$  die Matrix der Residuen ist, welche uns erst nach der Berechnung von  $\mathbf{Z}$  und  $\mathbf{B}$  zur Verfügung stehen. Im Folgenden sei durch  $\mathbf{X} = \mathbf{Z}\mathbf{B}^T$  eine volle Rang Approximation gegeben. Bei der klassischen Hauptkomponentenanalyse multipliziert man beide Seiten mit  $\mathbf{B}$ , um die Hauptkomponenten  $\mathbf{X}\mathbf{B} = \mathbf{Z}\mathbf{B}^T\mathbf{B} = \mathbf{Z}$  zu erhalten. Falls die Hauptachsen nicht orthogonal zueinander sind, ist der letzte Schritt nicht mehr gültig. Daher korrigiert man bei der dünnbesetzten Variante mit der Moore-Penrose-Inversen  $(\mathbf{B}^T\mathbf{B})^+$  ähnlich zu der Methode der kleinsten Quadrate. Demnach sollten die Hauptkomponenten durch

$$\mathbf{Z} = \mathbf{X}\mathbf{B}^T(\mathbf{B}^T\mathbf{B})^+ \quad (4.8)$$

berechnet werden. Falls  $\mathbf{B}$  orthogonal ist, reduziert sich (4.8) wie gehabt auf  $\mathbf{Z} = \mathbf{X}\mathbf{B}$ . Ein weiterer Unterschied ist, dass nun keine sequentielle Berechnung der Hauptkomponenten mehr möglich ist. Jede Hauptkomponente hängt durch (4.8) von allen Hauptachsen ab, weshalb sich die Werte bei Hinzunahme weiterer Hauptachsen zum Modell jedes mal ändern können. Camacho et al. zeigen empirisch, dass die Korrelation zwischen den Hauptkomponenten für der von uns betrachteten Variante der dünnbesetzten Hauptkomponentenanalyse deutlich sinkt, was eine Interpretation weiter verbessert.

Für die Modellbewertung wird oft die erklärte Varianz des Datensatzes herangezogen, wie in Abschnitt 3.2 beschrieben. Zou et al. erkennen, dass aufgrund der Korrelation der Hauptkomponenten die Varianzen nicht wie gewohnt errechnet werden können und schlagen in [ZHT06] folgende Methode vor. Die erklärte Varianz für die ersten  $j + 1$  Hauptkomponenten sollte sich aus der Summe der ersten  $j$  zusammen mit der erklärten Varianz der  $k + 1$ -ten Hauptkomponente  $Z_{j+1}$  ergeben. Aufgrund der Korrelation der Hauptkomponenten erhält die Varianz von  $Z_{j+1}$  aber Beiträge anderer Hauptkomponenten. Um nur die zusätzlich durch  $Z_{j+1}$  erhaltene Varianz zu erhalten und lineare Abhängigkeiten zu entfernen, nutzen Zou et al. eine Projektion

$$Z_{j \cdot 1, \dots, j-1} = Z_j - \mathbf{P}_{1, \dots, j-1} Z_j \quad (4.9)$$

wobei  $\mathbf{P}_{1, \dots, j-1}$  die orthogonale Projektionsmatrix auf  $\{Z_i\}_1^{j-1}$  sei. Mit  $Z_{j \cdot 1, \dots, j-1}$  bezeichnen wir also die Residuen nach Anpassung von  $Z_j$  durch  $Z_1, \dots, Z_{j-1}$ . Man beachte, dass (4.9) von der Reihenfolge der  $Z_i$  abhängt. Aufgrund der natürlichen Ordnung bei der Hauptkomponentenanalyse stellt dies aber kein Problem dar. Somit ergibt sich die Gesamtvarianz der ersten  $k$  Hauptkomponenten durch

$$\sum_{j=1}^k \|Z_{j \cdot 1, \dots, j-1}\|^2. \quad (4.10)$$

Mithilfe einer QR-Zerlegung von  $\mathbf{Z} = \mathbf{Q}\mathbf{R}$ , wobei  $\mathbf{Q}$  orthonormal und  $\mathbf{R}$  eine recht obere Dreiecksmatrix ist, können wir (4.10) schnell berechnen, denn  $\|Z_{j \cdot 1, \dots, j-1}\|^2 = R_{jj}^2$ . Auch wenn dieser Ansatz zunächst sinnvoll scheinen mag, werden wir in Kapitel 6 anhand unseres Datensatzes zeigen, dass durch (4.10) keine korrekte Berechnung der erklärten Varianz erfolgt. Das Problem des Ansatzes liegt daran, dass der Zusammenhang zum Rekonstruktionsfehler nicht klar ist. Bei der klassischen Hauptkomponentenanalyse haben wir die erklärte Varianz mithilfe des Rekonstruktionsfehlers angeben können, was hier nicht mehr der Fall sein muss.

Eine korrekte Methode wird von Camacho et al. eingeführt. Hierbei zerlegen wir die Varianz des Modells  $\mathbf{X} = \mathbf{Z}\mathbf{B}^T + \mathbf{E}$  in zwei Teile.

$$\begin{aligned} \|\mathbf{X}\|_F^2 &= \|\mathbf{Z}\mathbf{B}^T + \mathbf{E}\|_F^2 \\ &= \text{tr}(\mathbf{B}\mathbf{Z}^T\mathbf{Z}\mathbf{B}^T) + \text{tr}(\mathbf{B}\mathbf{Z}^T\mathbf{E}) + \text{tr}(\mathbf{E}^T\mathbf{Z}\mathbf{B}^T) + \text{tr}(\mathbf{E}^T\mathbf{E}) \\ &= \text{tr}(\mathbf{B}\mathbf{Z}^T\mathbf{Z}\mathbf{B}^T) + \text{tr}(\mathbf{E}^T\mathbf{E}) \\ &= \|\mathbf{Z}\mathbf{B}^T\|_F^2 + \|\mathbf{E}\|_F^2 \end{aligned} \quad (4.11)$$

If deflation is performed in the score space  $EP = P^TE^T = 0$ . Damit kann die Varianz eines Datensatzes in die Varianz der Rekonstruktion und der Residuen aufgeteilt werden. Ersterer Teil entspricht der erklärten Varianz unseres Modells und wird daher mit  $\|\mathbf{Z}\mathbf{B}^T\|_F^2$  berechnet.

## 4.5 Wahl der Hyperparameter

Bei der dünnbesetzten Hauptkomponentenanalyse sind mehrere Hyperparameter zu wählen. Dazu gehören die Anzahl an Hauptkomponenten  $k$  und die Regularisierungsparameter  $\lambda_{1,j}$  und  $\lambda_2$ . Im Folgenden möchte wir mögliche Vorgehensweise näher erläutern und eine Übersicht über mögliche Verfahren geben.

Bevor man die Regularisierungsparameter festlegt ist es sinnvoll zunächst die Anzahl an Hauptkomponenten für das Modell zu bestimmen, da sich bei einer Änderung von  $k$  alle Hauptkomponenten verändern können. Hierbei kann man analog zur klassischen Variante wie in Abschnitt 3.2 beschrieben vorgehen. Man nutzt also zunächst die klassische Variante, um  $k$  zu bestimmen.

Empirische Ergebnisse zeigen, dass die Ergebnisse sich mit Veränderung von  $\lambda_2$  kaum ändern. Für einen  $n > p$  Datensatz kann der Parameter auf Null gesetzt werden, da das Lasso-Defizit in diesem Fall nicht auftritt. In der Praxis wird  $\lambda_2$  auf eine kleine positive Zahl in der Größenordnung  $10^{-6}$  gesetzt, um mögliche Kollinearitätsprobleme zu vermeiden [ZHT06]. Falls  $p \gg n$  werden wir eine Standardwahl von  $\lambda_2$  in Kapitel 5 treffen.

Komplizierter gestaltet sich eine Wahl von  $\lambda_{1,j}$ , welche die Balance zwischen Dünnbesetzung und Rekonstruktionsfehler regelt. Es wird keine Sturres Ausprobieren in Zou et al. Alternativ wird vorgeschlagen, den zweischrittigen Prozess aus REF zu verwenden, bei welcher man einen gesamten Lösungspfad für die  $\lambda_{i,j}$  erhält. Bei der Verwendung dieses Ansatzes kommen allerdings unterschiedliche Lösungsansätze.

Im Prinzip könnte man die  $\lambda_{1,j}$  durch ein Kreuzvalidierungsverfahren bestimmen. Je nach Größe des Datensatzes kann dies aber sehr rechenintensiv sein, weshalb wir hier einen alternativen Ansatz beschreiben möchten. In der Literatur wird meist ein Bayes-Informationskriterium (BIC) angegeben, welches aber je nach Anwendung und Generalisierung verschieden formuliert wird. Zwei Varianten, welche auf [Hub+16; AM11] (4.12) und [CFF13; Guo+10] (4.13) zurückgehen, werden wir hier betrachten.

$$\text{BIC}(\lambda_{1,j}) = \log \left( \frac{\|X - Z_j \beta_j^T\|_F^2}{np} \right) + \text{df}(\lambda_{1,j}) \frac{\log(np)}{np} \quad (4.12)$$

wobei  $\text{df}(\lambda_{1,j}) = \|\beta_j\|_0$  die Anzahl von null verschiedenen Einträge sind. Dabei steht  $\text{df}$  für *degrees of freedom*, welches die Anzahl freier Parameter bzw. den Freiheitsgrad darstellt [HTF09]. Klar erkennbar in (4.12) ist der Kompromiss zwischen Rekonstruktionsfehler der  $j$ -ten Hauptkomponente und der Dünnbesetzung durch  $\text{df}(\lambda_{1,j})$ . Mit steigendem  $\lambda_{1,j}$  wird der Rekonstruktionsfehler größer und die Anzahl freier Parameter geringer. Wir sind gehalten  $\lambda_{1,j}$  zu finden, die eine Balance zwischen den beiden Termen ermöglicht.

Um nicht über  $k$  Parameter optimieren zu müssen, kann man beispielsweise  $\lambda_{1,j} = \lambda_1$  für alle  $1 \leq j \leq k$  setzen. Eine weitere Möglichkeit wird in [CFF13] beschrieben. Für  $j > 1$  sei  $\tilde{\mathbf{B}}_{j-1}^\perp$  die Matrix, deren Spalten eine orthonormale Basis für das orthogonale Komplement für den durch  $\beta_1, \dots, \beta_{j-1}$  aufgespannten Raum sind. Wir bezeichnen  $x_i^{(j-1)} = (\tilde{\mathbf{B}}_{j-1}^\perp)^T x_i$  für  $i = 1, \dots, n$  und setzen  $\lambda_{1,j} = \lambda_1 \text{Var}[\mathbf{X}^{(j)}]$ , wobei  $\mathbf{X}^{(j)}$  aus den auf das orthogonale Komplement der ersten  $j-1$  Hauptachsen projizierten Daten  $x_i^{(j-1)}$  besteht. Mithilfe dieses Ansatzes können wir eine vergleichbare Dünnbesetzung für alle Hauptachsen erhalten. Für die Wahl von  $\lambda_1$  wird ein ähnliches BIC-Kriterium vorgeschlagen.

$$\text{BIC}(\lambda_1) = \frac{\|\mathbf{X} - \mathbf{XBA}^T\|_F^2}{\|\mathbf{X} - \mathbf{XVV}^T\|_F^2} + \text{df}(\lambda_1) \frac{\log(n)}{n} \quad (4.13)$$

Für die log-likelihood-Funktion wird in (4.13) das Verhältnis zwischen Rekonstruktionsfehler der dünnbesetzten und der klassischen Hauptkomponentenanalyse gewählt. Welche der beiden BIC-Kriterien genutzt werden sollte, kommt auf den Anwendungsfall und den Rechenaufwand an, welchen man bereit ist in Kauf zu nehmen.

Typischerweise wird für die Minimierung der BIC-Kriterien eine Rastersuche für  $\lambda_1$  im Wertebereich  $[0, \lambda_1^{max}]$  durchgeführt, wobei eine Wahl von  $\lambda_1^{max}$  in Hauptachsen mit nur einem von Null verschiedenem Eintrag resultieren. Andere Verfahren wie die Zufallssuche, Bayesche oder gradienbasierte Optimierung sind an dieser Stelle denkbar, um ein passendes  $\lambda_1$  zu finden.

Wichtig zu erwähnen ist, dass BIC-Kriterien sich in  $p \gg n$ -Situationen schlecht verhalten. The BIC suffers from two main limitations[5] the above approximation is only valid for sample size  $n$  much larger than the number  $k$  of parameters in the model. the BIC cannot handle complex collections of models as in the variable selection (or feature selection) problem in high-dimension.

## 4.6 Theoretische Aussagen

Zu Abschluss dieses Kapitels möchten wir uns mit theoretischen Aussagen zur dünnbesetzten Hauptkomponentenanalyse auseinandersetzen. Von wesentlicher Bedeutung ist die Konsistenz der Methode im Vergleich zur klassischen Variante.

Durch ref werden wir sehen, dass das Sparse PCA Kriterium nur von der Kovarianzmatrix abhängt. Um eine Populationsversion der dünnbesetzten Hauptkomponente zu erhalten ersetzen wir  $\mathbf{X}^T \mathbf{X}$  schlicht durch die Kovarianzmatrix  $\Sigma$ .

recomputation necessary of principal components when changing number of pc's.

Alexandre d'Aspremont, Optimal Solutions for Sparse Principal Component Analysis We then use the same relaxation to derive sufficient conditions for global optimality of a solution, which can be tested in  $O(n^3)$  per pattern. We discuss applications in subset selection and sparse recovery and show on artificial examples and biological data that our algorithm does provide globally optimal solutions in many cases.



## Kapitel 5

# Implementierung

In diesem Kapitel werden wir einen Algorithmus beschreiben, der das Sparse PCA Kriterium (4.6) minimiert. Dazu werden wir uns zunächst mit dem allgemeinen Fall beschäftigen bevor wir den Fall  $n \ll p$  genauer betrachten, um eine effiziente Berechnung zu garantieren. Zu Schluss dieses Kapitels werden wir uns genauer mit der Laufzeit des Algorithmus auseinandersetzen.

### 5.1 Numerische Lösung

Zunächst möchten wir das Sparse PCA Kriterium erneut formulieren. Sei  $\mathbf{A}_{p \times k} = [\alpha_1, \dots, \alpha_k]$  und  $\mathbf{B}_{p \times k} = [\beta_1, \dots, \beta_k]$ . Wie zuvor bezeichnen wir mit  $x_i$  die  $i$ -te Zeile von  $\mathbf{X}$ . Dann lautet das Sparse PCA Kriterium

$$(\hat{\mathbf{A}}, \hat{\mathbf{B}}) = \arg \min_{\mathbf{A}, \mathbf{B}} \sum_{i=1}^n \left\| x_i - \mathbf{A} \mathbf{B}^T x_i \right\|_2^2 + \lambda \sum_{j=1}^k \left\| \beta_j \right\|_2^2 + \sum_{j=1}^k \lambda_{1,j} \left\| \beta_j \right\|_1$$

unter der Nebenbedingung, dass  $\mathbf{A}^T \mathbf{A} = I_{k \times k}$

Durch die Einführung der Matrix  $\mathbf{B}$  in das Kriterium in Kapitel 4 wird über die beiden Matrizen  $\mathbf{A}$  und  $\mathbf{B}$  minimiert. Man kann zeigen, dass es sich bei (...) um ein nicht-konvexes Optimierungsproblem handelt. (CITE) Allerdings stellt sich heraus, dass das Problem konvex für festes  $\mathbf{A}$  bzw. festes  $\mathbf{B}$  ist. (Überprüfung!!!) Daher liegt es nahe einen alternierenden Ansatz zu wählen, um das Problem numerisch zu lösen. Wir betrachten im Folgenden also zwei Optimierungsprobleme.

**B gegeben A:**

Wir wenden uns zunächst der Verlustfunktion zu. Hierfür sei  $\mathbf{A}_\perp \in \mathbb{R}^{p \times (p-k)}$  eine orthonormale Matrix, so dass  $[\mathbf{A}; \mathbf{A}_\perp]$   $p \times p$  orthonormal ist. Dann gilt

$$\begin{aligned} \sum_{i=1}^n \left\| x_i - \mathbf{A} \mathbf{B}^T x_i \right\|_2^2 &= \left\| \mathbf{X} - \mathbf{X} \mathbf{B} \mathbf{A}^T \right\|_F^2 \\ &= \left\| \mathbf{X} \mathbf{A}_\perp \right\|_F^2 + \left\| \mathbf{X} \mathbf{A} - \mathbf{X} \mathbf{B} \right\|_F^2 \\ &= \left\| \mathbf{X} \mathbf{A}_\perp \right\|_F^2 + \sum_{j=1}^k \left\| \mathbf{X} \alpha_j - \mathbf{X} \beta_j \right\|_2^2 \end{aligned}$$

Wir setzen  $Y_j^* = \mathbf{X}\alpha_j$  für alle  $1 \leq j \leq n$ . Somit reduziert sich (...) für fixes  $\mathbf{A}$  auf das Lösen von  $k$  elastic net Problemen

$$\hat{\beta}_j = \arg \min_{\beta_j} \left\| Y_j^* - \mathbf{X}\beta_j \right\|_2^2 + \lambda_2 \|\beta_j\|_2^2 + \lambda_{1,j} \|\beta_j\|_1 \quad (5.1)$$

In Kapitel 2 haben wir uns bereits mit elastic nets beschäftigt und einen effizienten Algorithmus zur Lösung dieser präsentiert.

**A gegeben B:**

Fixieren wir die Matrix  $\mathbf{B}$ , so können wir uns auf das Minimieren der Zielfunktion  $\sum_{i=1}^n \|x_i - \mathbf{A}\mathbf{B}^T x_i\|_2^2 = \|\mathbf{X} - \mathbf{X}\mathbf{B}\mathbf{A}^T\|_F^2$  beschränken, da die Bedingungen an  $\beta_j$  nicht von Relevanz beim Optimieren über  $\mathbf{A}$  sind. Somit ergibt sich

$$\hat{\mathbf{A}} = \arg \min_{\mathbf{A}} \left\| \mathbf{X} - \mathbf{X}\mathbf{B}\mathbf{A}^T \right\|_F^2 \quad (5.2)$$

unter der Nebenbedingung, dass  $\mathbf{A}\mathbf{A}^T = \mathbf{I}_{k \times k}$

Für dieses Optimierungsproblem lässt sich eine explizite Lösung angeben, da es eine Form von Procrustes Rotationsproblem ist, welches wir in Theorem REF beschrieben haben. Sei daher

$$(\mathbf{X}^T \mathbf{X})\mathbf{B} = \mathbf{U}\mathbf{D}\mathbf{V}^T$$

eine Singulärwertzerlegung. Dann ist  $\hat{\mathbf{A}} = \mathbf{U}\mathbf{V}^T$ .

Es ist sinnvoll zu erwähnen, dass für die Lösung beider Teilprobleme nur die Gram-Matrix  $\mathbf{X}^T \mathbf{X}$  bekannt sein muss. Dies erleichtert die Berechnung.

## 5.2 Algorithmus

Durch die Vorarbeit im vorangegangenen Abschnitt können wir einen effizienten Algorithmus zur Lösung des Sparse PCA Kriteriums angeben. Zuerst initialisieren wir  $\mathbf{A}$  mit den ersten  $k$  Hauptachsen. Anschließend minimieren wir abwechselnd über  $\mathbf{B}$  gegeben  $\mathbf{A}$  und  $\mathbf{A}$  gegeben  $\mathbf{B}$  solange bis ein geeignetes Konvergenzkriterium erfüllt ist oder wir eine maximale Anzahl an Iterationen erreicht haben. Durch abschließende Normalisierung der Spalten von  $\mathbf{B}$  erhalten wir die dünnbesetzten Hauptachsen. Eine Übersicht haben wir in Algorithmus 1 erstellt.

Es stellt sich nun die Frage nach einem passendem Abbruchkriterium. Da für uns am Schluss des Algorithmus nur die dünnbesetzten Hauptachsen relevant sind, liegt es nahe ein Konvergenzkriterium für  $\mathbf{B}$  zu wählen. Zou et al. brechen in ihrer Implementierung die Iteration ab, falls

$$\max_{\substack{1 \leq i \leq p \\ 1 \leq j \leq k}} \left| \frac{\beta_{ij}^{(l+1)}}{\|\beta_i\|} - \frac{\beta_{ij}^{(l)}}{\|\beta_i\|} \right| < \epsilon$$

wobei  $\beta_{ij}^{(l)}$  der  $j$ -te Eintrag der dünnbesetzten Hauptachse  $\beta_i$  in der  $l$ -ten Iteration ist. Sobald also die Änderung in  $\mathbf{B}^{(l)}$  klein genug ist, kann die while-Schleife abgebrochen werden. Um die Laufzeit des Algorithmus zu beschränken, ist es sinnvoll ein zusätzliches Abbruchkriterium zu definieren. So werden wir bei Anwendung des Algorithmus eine maximale Anzahl an Iterationen  $l_{max}$  festlegen, die nicht überschritten werden darf.



**Algorithm 1** Sparse Principal Component Analysis

---

```

1: procedure SPCA(A, B)
2:   A  $\leftarrow$  V[, 1 : k], die ersten k Hauptachsen
3:   while nicht konvergiert do
4:     Gegeben festes A = [ $\alpha_1, \dots, \alpha_k$ ], löse das elastic net Problem
       
$$\hat{\beta}_j = \arg \min_{\beta_j} \|\mathbf{X}\alpha_j - \mathbf{X}\beta_j\|^2 + \lambda \|\beta_j\|_2^2 + \lambda_{1,j} \|\beta_j\|_1 \quad \text{für } j = 1, \dots, k$$

5:     Gegeben festes B = [ $\beta_1, \dots, \beta_k$ ], berechne die Singulärwerzerlegung von
       
$$\mathbf{X}^T \mathbf{X} \mathbf{B} = \mathbf{U} \mathbf{D} \mathbf{V}^T$$

       
$$\mathbf{A} \leftarrow \mathbf{U} \mathbf{V}^T$$

6:   end while
7:    $\hat{\mathbf{V}}_j \leftarrow \frac{\beta_j}{\|\beta_j\|}$  for  $j = 1, \dots, k$ 
8: end procedure

```

---

### 5.3 Komplexität

Wir werden uns nun mit der Komplexität von Algorithmus 1 beschäftigen. Dabei unterscheiden wir zwischen den beiden Fällen  $n > p$  und  $p \gg n$ , in welcher (5.1) auf eine andere Weise gelöst werden kann. Falls  $p$  nur geringfügig größer ist als  $n$  wenden wir weiterhin die beschriebene Methode an.

**Fall:  $n > p$** 

Für die Berechnung von (5.1) lässt sich ein Trick anwenden. Indem wir

$$\begin{aligned}
\hat{\beta}_j &= \arg \min_{\beta_j} \left\| Y_j^* - \mathbf{X}\beta_j \right\|_2^2 + \lambda \|\beta_j\|_2^2 + \lambda_{1,j} \|\beta_j\|_1 \\
&= \arg \min_{\beta_j} (\alpha_j - \beta_j)^T \mathbf{X}^T \mathbf{X} (\alpha_j - \beta_j) + \lambda_2 \|\beta_j\|_2^2 + \lambda_{1,j} \|\beta_j\|_1
\end{aligned} \tag{5.3}$$

umformen, sehen wir, dass sowohl (5.1) als auch (5.2) nur von der Kovarianzmatrix  $\mathbf{X}^T \mathbf{X}$  abhängen. Daher ist es sinnvoll, diese vorab zu berechnen, um die Anzahl an Multiplikation je Iteration zu verringern. Ersetzen wir  $\mathbf{X}^T \mathbf{X}$  durch die Kovarianzmatrix  $\mathbf{\Sigma}$ , erhalten wir eine Populationsversion der dünnbesetzten Hauptkomponentenanalyse. Auch wenn (5.3) mit  $\mathbf{\Sigma}$  nicht direkt ein Elastic Net Problem ist, kann man dies mithilfe von  $Y^{**} = \mathbf{\Sigma}^{\frac{1}{2}} \alpha_j$  und  $\mathbf{X}^{**} = \mathbf{\Sigma}^{\frac{1}{2}}$  in ein Elastic Net Problem transformieren

$$\hat{\beta}_j = \arg \min_{\beta} \|\mathbf{Y}^{**} - \mathbf{X}^{**} \beta\|_2^2 + \lambda \|\beta\|_2^2 + \lambda_{1,j} \|\beta\|_1$$

Dadurch ergibt sich die folgende Laufzeit für die einzelnen Komponenten in Algorithmus 1.

Berechnung	Komplexität
$\mathbf{X}^T \mathbf{X}$	$\mathcal{O}(np^2)$
$\mathbf{X}^T \mathbf{X} \mathbf{B}$	$\mathcal{O}(p^2 k)$
Singulärwertzerlegung von $\mathbf{X}^T \mathbf{X} \mathbf{B}$	$\mathcal{O}(pk^2)$
Lösung des Elastic Net Problems (5.3)	$\mathcal{O}(p^3)$

Insgesamt ergibt sich aufgrund von  $k \leq p$  eine Laufzeit von  $(np^2) + m\mathcal{O}(p^3)$ , wobei  $m$  die Anzahl an Iterationen darstellt. Diese Schranke ist sehr gut im Hinblick auf eine hohe Beobachtungszahl, wird aber für hochdimensionale Daten (z.B.  $p > 100$ ) ineffizient.

#### Fall: $p \gg n$

Wir bezeichnen mit  $J$  die Anzahl von Null verschiedenen Einträge der Hauptachsen. Die meiste Zeit wird zur Lösung des Elastic Net Problems benötigt, welches in  $\mathcal{O}(pnJ + J^3)$ .

### 5.4 Numerische Lösung im Fall $p \gg n$

Für viele Anwendungen kann die Anzahl an Variablen die Anzahl an Beobachtungen deutlich übersteigen. In diesem Fall ist sehr rechenintensiv.

Wir beobachten, dass (4.1) für alle  $\lambda_2$  gilt. Es stellt sich heraus, dass für  $\lambda_2 \rightarrow \infty$  eine explizite Lösung des Minimierungsproblem existiert.

**Theorem 5.1.** Seien  $\hat{\mathbf{B}}_j(\lambda_2) = \frac{\hat{\beta}_j}{\|\hat{\beta}_j\|}$  die dünnbesetzten Hauptachsen aus (4.6). und  $(\tilde{\mathbf{A}}, \tilde{\mathbf{B}})$  die Lösung des Optimierungsproblems

$$(\tilde{\mathbf{A}}, \tilde{\mathbf{B}}) = \arg \min_{\mathbf{A}, \mathbf{B}} -2\text{tr}(\mathbf{A}^T \mathbf{X}^T \mathbf{X} \mathbf{B}) + \sum_{j=1}^k \|\beta_j\|_2^2 + \sum_{j=1}^k \lambda_{1,j} \|\beta_j\|_1 \quad (5.4)$$

unter der Nebenbedingung, dass  $\mathbf{A}^T \mathbf{A} = \mathbf{I}_{k \times k}$ .

Für  $\lambda_2 \rightarrow \infty$  konvergieren die dünnbesetzten Hauptachsen  $\hat{\mathbf{B}}_j(\lambda_2) \rightarrow \frac{\tilde{\beta}_j}{\|\tilde{\beta}_j\|}$ .

Daher können wir das vereinfachte Optimierungsproblem (5.4) benutzen, um die Elastic Net Probleme für den Spezialfall  $\lambda_2 = \infty$  zu lösen. Fixieren wir wie in Algorithmus 1 die Matrix  $\mathbf{A}$ , verbleiben wir mit dem Problem

$$\hat{\beta}_j = \arg \min_{\beta_j} -2\alpha_j^T (\mathbf{X}^T \mathbf{X}) \beta_j + \|\beta_j\|_2^2 + \lambda_{1,j} \|\beta_j\|_1. \quad (5.5)$$

Für (5.5) können wir eine explizite Lösung angeben

$$\hat{\beta}_j = \text{soft}_{\frac{\lambda_{1,j}}{2}}(\alpha_j^T \mathbf{X}^T \mathbf{X}) = \left( |\alpha_j^T \mathbf{X}^T \mathbf{X}| - \frac{\lambda_{1,j}}{2} \right)_+ \text{Sign}(\alpha_j^T \mathbf{X}^T \mathbf{X}) \quad (5.6)$$

wobei  $\text{soft}_\delta(x)$  der soft-thresholding Operator aus Abschnitt 2.3 ist. Daher ersetzen wir für den Fall  $p \gg n$  die Berechnung in Schritt 4 des Algorithmus 1 durch den Spezialfall (5.6). Für den hochdimensionalen Fall kann somit ein effizientes Verfahren gewährleistet werden. (Dann dauert jedes elastic net nur noch  $\mathcal{O}(pn)$ , wieso ist  $\lambda_2 \rightarrow \infty$  sinnvoll? Und warum wählen wir den Weg nicht für große  $p$ ?)

UST totally ignores the dependence between predictors and treats them as independent variables. Although this may be considered illegitimate, UST and its variants are used in other

methods such as significance analysis of microarrays (Tusher et al., 2001) and the nearest shrunken centroids classifier (Tibshirani et al., 2002), and have shown good empirical performance

## 5.5 Eigene Implementierung in Python

Momentan existieren Implementierungen der dünnbesetzten Hauptkomponentenanalyse in R und Python. Zou et al. stellen das `elasticnet` package mit einer `spca`-Funktion, welche auf ihrem Ansatz beruht, in der Programmiersprache R zur Verfügung. Für die Lösung des Subproblems (5.1) wird der LARS-EN Algorithmus gewählt, welche eine Erweiterung des LARS-Algorithmus für Elastic Nets ist [ZH05]. Dagegen bietet `scikit-learn` eine `SparsePCA`-Variante in Python, welche auf Jennatton et al. [jennatton] zurückgeht und einen leicht anderen Ansatz verfolgt.

Um ein genaues Verständnis der Ergebnisse zu garantieren, haben wir uns dazu entschieden, eine eigene Implementierung in Python vorzunehmen, die auf dem Ansatz von Zou et al. beruht. Es wurde kritisch überprüft, dass der von uns implementierte Code dieselben Ergebnissen wie der `spca`-Algorithmus aus dem `elasticnet` package liefert. Dazu haben wir den Pitprops Datensatz aus [ZHT06], welcher oft als Benchmark genutzt wird, und zusätzlich den eigenen Datensatz, welchen wir in Kapitel 6 beschreiben, verwendet. Gegenüber der Implementierung im `elasticnet` package haben wir zwei entscheidende Änderungen vorgenommen, welche die Laufzeit in der Praxis verkürzen. Statt das Subproblem (5.1) mit LARS-EN zu lösen, wählen wir ein Koordinaten-Abstiegsverfahren, welches wir in Abschnitt 2.3.5 beschrieben haben. Des Weiteren wird in der Implementierung von Zou et al. die Gram-Matrix  $\mathbf{X}^T \mathbf{X}$  vorab berechnet, um für das Subproblem (5.2) nur eine Multiplikation pro Iteration  $(\mathbf{X}^T \mathbf{X}) \mathbf{B}$  durchführen zu müssen. Da die Gram-Matrix aber in  $\mathbb{R}^{p \times p}$  liegt, ist es für den Fall  $p \gg n$  sinnvoller,  $\mathbf{X}^T (\mathbf{X} \mathbf{B})$  in jeder Iteration naiv zu berechnen, damit keine  $p \times p$ -Matrix zwischengespeichert werden muss. Dies ermöglicht für unseren hochdimensionalen Datensatz eine Laufzeit, die etwa um den Faktor 4 besser ist.

Bezüglich der Aufrufstruktur der Methode haben wir eine Reparametrisierung vorgenommen, um die Notation mit der `ElasticNet`-Klasse in `scikit-learn` zu vereinheitlichen. Ähnlich wie in Abschnitt 2.3.6 definieren wir

$$\alpha = \frac{2\lambda_2 + \lambda_1}{2n} \quad \text{und} \quad \gamma = \frac{\lambda_1}{2\lambda_2 + \lambda_1} \quad (5.7)$$

wobei  $\alpha$  die Stärke der Bestrafung und  $\gamma$  das Verhältnis des  $\ell_1$  zum  $\ell_2$ -Strafterm beschreibt.



## Kapitel 6

# Anwendung

In diesem Kapitel beschäftigen wir uns mit der Anwendung der dünnbesetzten Hauptkomponentenanalyse auf Frequenzdaten einer Maschine. Nicht unbedingt an der niedrigdimensionalen Repräsentation interessiert, sondern viel mehr an ...

Kurze Beschreibung der Idee, was wir uns erhoffen? Zunächst werden wir in Abschnitt 6.1 den Datensatz beschreiben und Vorverarbeitungsschritte erläutern.

Ob eine derartige Methode für diesen Datensatz sinnvoll war, werden wir in Kapitel 7 diskutieren.

### 6.1 Beschreibung des Datensatzes

grobe Einführung Messdaten akustischer und beschleunigungssensoren, welche die Vibration einer Maschine. (DARF ich hier die Idee nennen, warum die Messungen erhoben worden sind? Wir erhoffen bestimmte Frequenzen rauszufiltern, welche eine Aussage über die Partikelgröße ermöglichen. Was wurde geändert zwischen den Messungen? Rotation speed, flow pressure, und ... Sensoren an verschiedenen Teilen der Maschine angebracht, z.B. an der Mahlkammer, Auslassrohr oder am Sieb. Messungen mit und ohne Material, drei Messungen doppelt, 8 verschiedene Betriebszustände, einmal mit einmal ohne Material Maschinenfrequenzen rausfiltern.

Dadurch, dass die Messungen mit einer hohen Abtastrate durchgeführt worden sind, haben wir es mit einem hochdimensionalen Datensatz zu tun. Für jeden an der Maschine angebrachten Sensor haben wir somit  $p = 5120000$  Variablen. Mit nur  $n = 33$  Messungen, wobei Teile mehrmals aufgenommen worden sind, sind wir mit einer *high dimension low sample size* (HDLSS) Situation konfrontiert. Auf dieser Art Datensatz sind viele Methoden des unüberwachten Lernens unbrauchbar, weshalb eine Dimensionsreduktion sinnvoll sein kann. In diesem Fall sind wir aber weniger an der niedrigdimensionalen Repräsentation der Daten interessiert als an der Herausfilterung wichtiger Frequenzen, die eine Trennung der verschiedenen Messungen ermöglicht. Diese Möglichkeit der Interpretation war Anlass für die Verwendung der dünnbesetzten Hauptkomponentenanalyse. Des Weiteren haben wir in Abschnitt 3.5 theoretische Aussagen für eine Inkonsistenz der Hauptachsen in einer solchen Situation präsentiert. Mit der dünnbesetzten Variante können wir die Konsistenz wiederherstellen.

### 6.2 Vorverarbeitung der Daten

Vor der Anwendung der dünnbesetzten Hauptkomponentenanalyse auf den Datensatz haben wir einige Vorverarbeitungsschritte vorgenommen. Anfänglich sind uns Zeitreihen für

sowohl die Mikrofone als auch für die Beschleunigungssensoren gegeben. Da die Messungen zu zufälligen Zeitpunkten bei laufendem Mahlprozess gestartet worden sind, können einzelne Zeitpunkte nicht direkt miteinander verglichen werden. Somit haben wir eine Fouriertransformation der Daten angewandt. Hierdurch arbeiten wir anstatt auf der Zeitreihe also mit Frequenzdaten. In Abbildung REF zeigen wir beispielhaft das Ergebnis einer Fouriertransformation. Kurze Erklärung was wir auf x-y-Achse sehen. Dadurch erhoffen wir uns zusätzlich, dass Rauscheffekte besser erkennbar sind.

Es wird sich zeigen, dass der Algorithmus für die dünnbesetzte Hauptkomponentenanalyse sehr rechenintensiv sein kann. Daher haben wir uns entschieden nur einen Teil der ursprünglichen Zeitreihe zu verwenden. In Abbildung sieht man, dass sich die Frequenzen über die Zeit kaum ändern, was daran liegt, dass konstant Material in die Maschine zugeführt wird. Somit können wir die Dimension um einen Faktor 100 reduzieren ohne wichtige Informationen zu verlieren. Des Weiteren wurden Teile der Frequenzen, welche außerhalb des Frequenzbereichs des jeweiligen Sensors liegen abgeschnitten.

Ähnlich wie bei der klassischen Hauptkomponentenanalyse werden die Daten vor Anwendung zentriert, so dass die Varianz zwischen den Variablen vergleichbar ist. Eine Standardisierung haben wir nicht vorgenommen. Standardisieren ja/nein?

### 6.3 Anwendung auf Frequenzdaten

Unsere Implementierung ermöglicht es verschiedene Modellparameter zu wählen. Für eine Beschränkung der Laufzeit setzen wir eine maximale Anzahl an Iterationen von 500 und eine Toleranz von  $10^{-4}$ . Falls nach 500 Iterationen die vorgegebene Toleranz noch immer nicht erreicht ist, werden wir dies im Folgenden kenntlich machen. Ein weiterer Parameter den es zu wählen gilt, ist die Anzahl zu berechnender Hauptkomponenten. Wie bereits in ??beschrieben kann sich die Zusammensetzung der ersten Hauptachsen durchaus ändern, falls man weitere Hauptkomponenten berechnen möchte. In unserem Fall haben wir daher einen Durchlauf mit 2 bzw. 10 Hauptkomponenten gestartet. Interessanter jedoch ist die Wahl der Hyperparameter  $\lambda_1$  und  $\lambda_2$ , welche wesentlichen Einfluss auf die Ergebnisse besitzen.

Mithilfe eines BIC und AIC Kriteriums auswählen?

Exemplarisch werden wir uns nun mit einem der akustischen und einem der Vibrations-sensoren beschäftigen. An dieser Stelle möchten wir erwähnen, dass wir die Sensoren getrennt betrachten, d.h. die dünnbesetzte Hauptkomponentenanalyse einzeln auf die Sensoren anwenden. (Sonst Standardisierung aufgrund verschiedener Skalen notwendig, welche zu schlechteren Ergebnissen führt)

### 6.4 Auswertung der Ergebnisse

Vergleich Tabelle PCA / Sparse PCA (Loadings)

Exemplarisch zeigen, dass wir dünnbesetzte Hauptachsen haben. Klare Trennung z.B. nach rotation speed. Erklärte Varianz dropt.

Um einen Vergleich zu ermöglichen, haben wir zeitgleich die klassische Hauptkomponentenanalyse auf den Datensatz angewandt. Wir möchten nun ausgewählte Ergebnisse vorstellen. Zeitgleich werden wir die Resultate mit

Effekte: Recomputation necessary if dimension changing

## **6.5 Vergleich zur klassischen Hauptkomponentenanalyse**

## **6.6 Effekte der Hyperparameter**

Veränderung des Hyperparameters und dessen Effekte

### **6.6.1 Zeit**

evtl. Zeit pro Iteration betrachten?

### **6.6.2 Anzahl an Iterationen**

### **6.6.3 Effekt auf Resultate**





## Kapitel 7

# Ausblick / Zusammenfassung

### 7.1 Einsetzbarkeit

Wann ist die Methode sinnvoll einzusetzen?

We now have a good understanding of the role of sparsity in PCA and ways to effectively exploit the sparsity. There are still remaining issues. A very important issue to be investigated further is automated SPCA. By “automated” we mean that there is a principled but not overly complicated procedure to set these sparse parameters in SPCA. This question is particularly challenging when we solve several sparse principal components jointly.

Kritik results in very large residuals highly correlated scores  $q$  is no updated due to the normalization step

Subproblem schneller lösbar mit Verbindung zu SVM

### 7.2 Übertragbarkeit

Übertragbarkeit auf andere Datensätze

### 7.3 Ongoing Research / Weitere Techniken



# Literatur

- [AM11] Genevera I Allen und Mirjana Maletić-Savatić. „Sparse non-negative generalized PCA with applications to metabolomics“. In: *Bioinformatics* 27.21 (2011), S. 3029–3035. URL: <https://doi.org/10.1093/bioinformatics/btr522>.
- [And03] Theodore W. Anderson. „An Introduction to Multivariate Statistical Analysis“. In: (2003).
- [Ant98] Howard Anton. *Lineare Algebra: Einführung, Grundlagen, Übungen*. Spektrum Akademischer Verlag, 1998. ISBN: 9783827403247.
- [Bis06] Christopher M Bishop. *Pattern recognition and machine learning*. 1. Aufl. Springer-Verlag New York, 2006. ISBN: 978-0-387-31073-2.
- [BSP12] Y. Murali Mohan Babu, M. V. Subramanyam und M. N. Giri Prasad. „PCA based image denoising“. In: 2012. URL: <https://doi.org/10.5121/sipij.2012.3218>.
- [Cam+20] J Camacho u. a. „All sparse PCA models are wrong, but some are useful. Part I: Computation of scores, residuals and explained variance“. In: *Chemometrics and Intelligent Laboratory Systems* 196 (2020), S. 103907. URL: <https://doi.org/10.1016/j.chemolab.2019.103907>.
- [Can+11] Emmanuel J Candès u. a. „Robust principal component analysis?“ In: *Journal of the ACM (JACM)* 58.3 (2011), S. 1–37. URL: <http://doi.acm.org/10.1145/1970392.1970395>.
- [CDS98] Scott Shaobing Chen, David L. Donoho und Michael A. Saunders. „Atomic Decomposition by Basis Pursuit“. In: *SIAM J. Sci. Comput.* 20.1 (1998), S. 33–61. URL: <https://doi.org/10.1137/S1064827596304010>.
- [CFF13] Christophe Croux, Peter Filzmoser und Heinrich Fritz. „Robust Sparse Principal Component Analysis“. In: *Technometrics* 55.2 (2013), S. 202–214. URL: <https://doi.org/10.1080/00401706.2012.727746>.
- [CJ95] Jorge Cadima und Ian T. Jolliffe. „Loading and correlations in the interpretation of principle compenents“. In: *Journal of Applied Statistics* 22.2 (1995), S. 203–214. URL: <https://doi.org/10.1080/757584614>.
- [dAs+07] Alexandre d’Aspremont u. a. „A Direct Formulation for Sparse PCA Using Semidefinite Programming“. In: *SIAM Review* 49.3 (2007), S. 434–448. URL: <http://www.jstor.org/stable/20453990>.
- [EY36] Carl Eckart und Gale Young. „The approximation of one matrix by another of lower rank“. In: *Psychometrika* 1.3 (1936), S. 211–218. URL: <https://doi.org/10.1007/BF02288367>.
- [FHT10] Jerome Friedman, Trevor Hastie und Rob Tibshirani. „Regularization paths for generalized linear models via coordinate descent“. In: *Journal of statistical software* 33.1 (2010), S. 1. URL: <https://doi.org/10.18637/jss.v033.i01>.
- [Fis13] Gerd Fischer. *Lineare Algebra: Eine Einführung für Studienanfänger*. 18. Aufl. Vieweg+Teubner Verlag, 2013. ISBN: 9783834893659.
- [FR13] Simon Foucart und Holger Rauhut. *A Mathematical Introduction to Compressive Sensing*. Bd. 1. Birkhäuser Basel, 2013.
- [GD+04] John C Gower, Garnt B Dijkstra u. a. *Procrustes problems*. Bd. 30. Oxford University Press on Demand, 2004.

- [GD09] Yue Guan und Jennifer G. Dy. „Sparse probabilistic principal component analysis“. In: Bd. 5. 2009, S. 185–192.
- [GD14] Matan Gavish und David L. Donoho. „The Optimal Hard Threshold for Singular Values is  $4/\sqrt{3}$ “. In: *IEEE Transactions on Information Theory* 60.8 (2014), S. 5040–5053. URL: <https://doi.org/10.1109/TIT.2014.2323359>.
- [GML03] Paul Geladi, Marena Manley und Torbjörn Lestander. „Scatter plotting in multivariate data analysis“. In: *Journal of Chemometrics: A Journal of the Chemometrics Society* 17.8-9 (2003), S. 503–511. URL: <https://doi.org/10.1002/cem.814>.
- [Guo+10] Jian Guo u. a. „Principal Component Analysis With Sparse Fused Loadings“. In: *Journal of Computational and Graphical Statistics* 19.4 (2010), S. 930–946. URL: <https://doi.org/10.1198/jcgs.2010.08127>.
- [Hie19] Matthias Hieber. „Analysis in metrischen Räumen“. In: *Analysis II*. Springer Berlin Heidelberg, 2019. ISBN: 978-3-662-57542-0.
- [HTF09] Trevor Hastie, Robert Tibshirani und Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Bd. 2. Springer-Verlag New York, 2009.
- [Hub+16] Mia Hubert u. a. „Sparse PCA for High-Dimensional Data With Outliers“. In: *Technometrics* 58.4 (2016), S. 424–434. URL: <https://doi.org/10.1080/00401706.2015.1093962>.
- [Jän07] Klaus Jänich. *Lineare Algebra*. 11. Aufl. Springer-Lehrbuch. Springer Berlin Heidelberg, 2007. ISBN: 9783540755029.
- [JM09] Sungkyu Jung und J Stephen Marron. „PCA consistency in high dimension, low sample size context“. In: *The Annals of Statistics* 37.6B (2009), S. 4104–4130. URL: <https://doi.org/10.1214/09-AOS7090>.
- [Jou+10] Michel Journée u. a. „Generalized power method for sparse principal component analysis“. In: *Journal of Machine Learning Research* 11.Feb (2010), S. 517–553. URL: <https://dl.acm.org/doi/10.5555/1756006.17560210>.
- [JTU03] Ian T. Jolliffe, Nickolay T. Trendafilov und Mudassir Uddin. „A Modified Principal Component Technique Based on the LASSO“. In: *Journal of Computational and Graphical Statistics* 12.3 (2003), S. 531–547. URL: <https://doi.org/10.1198/1061860032148>.
- [JY09] Iain M. Johnstone und Arthur Yu Lu. „On Consistency and Sparsity for Principal Components Analysis in High Dimensions“. In: *Journal of the American Statistical Association* 104.486 (2009), S. 682–693. URL: <https://doi.org/10.1198/jasa.2009.0121>.
- [Koh05] Wolfgang Kohn. *Statistik: Datenanalyse und Wahrscheinlichkeitsrechnung*. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg, 2005. ISBN: 978-3-540-26768-3.
- [Kri+08] Hans-Peter Kriegel u. a. „A General Framework for Increasing the Robustness of PCA-Based Correlation Clustering Algorithms“. In: *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2008, S. 418–435. URL: [https://doi.org/10.1007/978-3-540-69497-7\\_27](https://doi.org/10.1007/978-3-540-69497-7_27).
- [LV07] John A. Lee und Michel Verleysen. *Nonlinear Dimensionality Reduction*. Springer New York, 2007. ISBN: 978-0-387-39351-3. URL: <https://doi.org/10.1007/978-0-387-39351-3>.
- [Mur12] Kevin P Murphy. *Machine learning: A Probabilistic Perspective*. Cambridge, MA: MIT press, 2012.
- [MWA06] Baback Moghaddam, Yair Weiss und Shai Avidan. „Spectral Bounds for Sparse PCA: Exact and Greedy Algorithms“. In: *Advances in Neural Information Processing Systems* 18. Hrsg. von Y. Weiss, B. Schölkopf und J. C. Platt. MIT Press, 2006, S. 915–922. URL: <http://papers.nips.cc/paper/2780-spectral-bounds-for-sparse-pca-exact-and-greedy-algorithms.pdf>.

- [Pau07] Debashis Paul. „Asymptotics of sample eigenstructure for a large dimensional spiked covariance model“. In: *Statistica Sinica* (2007), S. 1617–1642. URL: <https://www.jstor.org/stable/24307692>.
- [Ped+11] F. Pedregosa u. a. „Scikit-learn: Machine Learning in Python“. In: *Journal of Machine Learning Research* 12 (2011), S. 2825–2830.
- [Ric12] Richtárik, Peter and Takáč, Martin and Ahipasaoglu, Selin Damla. *Alternating Maximization: Unifying Framework for 8 Sparse PCA Formulations and Efficient Parallel Codes*. 2012. URL: <https://arxiv.org/abs/1212.4137>.
- [Rüs14] Ludger Rüschendorf. *Mathematische Statistik*. Berlin: Springer Spektrum, 2014. ISBN: 978-3-642-41996-6.
- [SH08] Haipeng Shen und Jianhua Z Huang. „Sparse principal component analysis via regularized low rank matrix approximation“. In: *Journal of multivariate analysis* 99.6 (2008), S. 1015–1034. URL: <https://doi.org/10.1016/j.jmva.2007.06.007>.
- [SSM13] Dan Shen, Haipeng Shen und James Stephen Marron. „Consistency of sparse PCA in high dimension, low sample size contexts“. In: *Journal of Multivariate Analysis* 115 (2013), S. 317–333.
- [SW04] Robert Schaback und Holger Wendland. *Numerische Mathematik*. 5. Aufl. Mathematics and Statistics. Springer, 2004. ISBN: 9783540213949.
- [Tai+17] Jiang Tai-Xiang u. a. „Patch-Based Principal Component Analysis for Face Recognition“. In: *Computational Intelligence and Neuroscience* (2017). URL: <https://doi.org/10.1155/2017/5317850>.
- [Tib+] Robert Tibshirani u. a. *Diabetes Data*. <https://www4.stat.ncsu.edu/~boos/var.select/diabetes.html>. Zugriff: 20. Januar 2020.
- [Tib+04] Robert Tibshirani u. a. „Least angle regression“. In: *The Annals of Statistics* 32.2 (2004), S. 407–499. URL: <http://dx.doi.org/10.1214/009053604000000067>.
- [Tib96] Robert Tibshirani. „Regression Shrinkage and Selection via the Lasso“. In: *Journal of the Royal Statistical Society. Series B (Methodological)* 58.1 (1996), S. 267–288. URL: <http://www.jstor.org/stable/2346178>.
- [TTT99] Kim-Chuan Toh, Michael J Todd und Reha H Tütüncü. „SDPT3 — a MATLAB software package for semidefinite programming“. In: *Optimization methods and software* 11.1-4 (1999), S. 545–581. URL: <https://doi.org/10.1080/10556789908805762>.
- [Van19] L. Vandenberghe. „Optimization Methods for Large-Scale Systems“. In: *Lecture Notes ECE236C, UCLA* (2019). URL: <http://www.seas.ucla.edu/~vandenbe/ee236c.html>.
- [VMS16] R. Vidal, Y. Ma und S. Sastry. *Generalized Principal Component Analysis*. Bd. 1. Interdisciplinary Applied Mathematics. Springer New York, 2016. ISBN: 9780387878119. URL: <https://doi.org/10.1007/978-0-387-87811-9>.
- [WTH09] Daniela M Witten, Robert Tibshirani und Trevor Hastie. „A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis“. In: *Biostatistics* 10.3 (2009), S. 515–534. URL: <https://doi.org/10.1093/biostatistics/kxp008>.
- [Yan+13] Allen Y. Yang u. a. „Fast  $\ell_1$ -Minimization Algorithms for Robust Face Recognition“. In: *IEEE Transactions on Image Processing* 22.8 (Aug. 2013), S. 3234–3246. ISSN: 1941-0042. URL: <http://dx.doi.org/10.1109/TIP.2013.2262292>.
- [YL06] Ming Yuan und Yi Lin. „Model selection and estimation in regression with grouped variables“. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 68.1 (2006), S. 49–67.
- [ZH05] Hui Zou und Trevor Hastie. „Regularization and Variable Selection via the Elastic Net“. In: *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* 67.2 (2005), S. 301–320. URL: <http://www.jstor.org/stable/3647580>.

- [ZHT06] Hui Zou, Trevor Hastie und Robert Tibshirani. „Sparse Principal Component Analysis“. In: *Journal of Computational and Graphical Statistics* 15.2 (2006), S. 265–286. URL: <https://doi.org/10.1198/106186006X113430>.