Analyse dünnbesetzter Hauptachsen für Frequenzdaten

Tobias Bork

Geboren am 21. November 1997 in Reutlingen

24. 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

An dieser Stelle möchte ich mich bei Prof. Dr. Jochen Garcke für die Möglichkeit bedanken, meine Bachelorarbeit mit Anwendung auf einen realen Datensatz schreiben zu dürfen. Des Weiteren danke ich Dr. X Y für die Übernahme der Zweitkorrektur. Dank geht auch an meine beiden Betreuer Dr. Sebastian Mayer und Christian Gscheidle, welche mir bei mathematischen wie physikalischen Fragen zur Seite standen und sich immer für mich Zeit genommen haben.

Für die nötige Motivation im Studium, die gemeinsame Zeit und nicht zuletzt auch für das Korrekturlesen dieser Arbeit bedanke ich mich bei Hendrik Baers, Christopher Reiners und Lennard Schiefelbein.

Zudem bedanke ich mich bei meiner Familie und Maja Peters für die reichlichen Gespräche und Ratschläge während des Studiums. Die bedingungslose Unterstützung hat mir in vielen Situation geholfen, die richtigen Entscheidungen zu treffen.

Inhaltsverzeichnis

Kapitel 1

Einführung

"We are drowning in information but starved for knowledge."

– John Naisbitt

Wir befinden uns in einem Zeitalter, in welchem Tag für Tag mehr Daten generiert werden. Im Jahr 2019 wurden pro Minute mehr als 500,000 Tweets auf Twitter gesendet, mehr als 250,000 Stories auf Instagram gepostet und ungefähr 4.5 Millionen Suchanfragen bei Google ausgeführt [DOM19]. Ein Rückgang dieser Entwicklung ist dabei nicht in Sicht. Eher im Gegenteil wurden mehr als 90% der Daten auf der Welt in den letzten zwei Jahren (Stand: 2017) erzeugt [IBM17]. Das exponentielle Datenwachstum erfordert die Entwicklung und Analyse von modernen Verfahren, welche mit dieser Masse an Daten umgehen können. *Big Data* ist ohne solcher Methoden nutzlos.

Motivation

Das Objekt von Interesse ist oft in riesigen Datensätzen verborgen bzw. nur von Teilen davon abhängig. In diesem Zusammenhang können Computer helfen, relevante Informationen zu extrahieren. Dies verlangt eine Identifizierung der signifikanten Variablen und das Verständnis deren Zusammenspiels. Im Bereich des maschinellen Lernens ergeben sich häufig zwei Ziele. Einerseits möchte man einen bestehenden Trainingsdatensatz genauer verstehen, strukturieren und vereinfachen. Andererseits interessiert man sich dafür, die erlernten Gesetzmäßigkeiten des Datensatzes auf neue, ungesehene Testdaten zu verallgemeinern.

Für dieser Art Aufgaben sind bereits viele effiziente Algorithmen entwickelt worden. Hochdimensionale Datensätze, in welche eine Vielzahl an Variablen einfließen, können solche Methoden aber häufig vor Probleme stellen. Dieses Phänomen ist unter dem dem sog. *Fluch der Dimensionen* bekannt, welcher sich in vielen verschiedenen Facetten niederschlägt [Bel61]. Ein offensichtliches Problem ist die Visualisierung von Datensätzen mit mehr als drei Dimensionen. Wir Menschen können Zusammenhänge in einem anschaulichen Kontext viel besser verstehen als in einer Tabelle voller Zahlen. Viele Eigenschaften des zwei- oder dreidimensionalen Raums lassen sich nicht auf höherdimensionale Räume übertragen. Im Zuge einer explorativen Datenanalyse kann sich die Projektion in niedrigdimensionale Räume daher als nützlich erweisen.

Ein weiterer Gesichtspunkt beschäftigt sich mit der Verallgemeinerung des Modells auf neue Daten. Ohne vereinfachende Annahmen muss die Anzahl an Beobachtungen exponentiell im Vergleich zu der Anzahl an Dimensionen wachsen, um weiterhin akkurate Vorhersagen zu treffen. In vielen Situationen verfügen wir zwar über eine Vielzahl an Variablen, aber nicht über genügend Beobachtungen, da die Beschaffung unter Umständen hohe Kosten verursachen kann.

Dimensionsreduktionsverfahren

Um den Fluch der Dimensionen zu umgehen, können Dimensionsreduktionsverfahren hilfreich sein. Sie zielen darauf ab, Daten in eine einfachere Repräsentation zu übersetzen, welche die Struktur trotz drastischer Reduktion möglichst genau wiedergibt. Dadurch können hochdimensionale Datensätze veranschaulicht und für weitere Analysezwecke verwendet werden. Anwendung finden Dimensionsreduktionsverfahren vor allem in der Bildverarbeitung, der Biologie und der Ingenieurwissenschaft. Einige interessante Beispiele beinhalten die Klassifizierung handgeschriebener Zahlen [HTF09], die Gesichtserkennung [HBB96] oder die Genexpressionsanalyse [ABB00].

Allgemein fallen Dimensionsreduktionsverfahren in zwei verschiedene Klassen, *feature selection* und *feature extraction*. Erstere erreichen eine Reduktion vor allem durch eine Filterung nach relevanten Variablen. Anstatt unwichtige Variablen zu vernachlässigen, beabsichtigen feature extraction Verfahren die Ausgangsvariablen zu kombinieren und zusammenzufassen. Sowohl die klassische Hauptkomponentenanalyse als auch die dünnbesetzte Variante, welche im Fokus dieser Arbeit steht, fallen unter diese Kategorie.

Transparenz des Modells

Modelle der künstlichen Intelligenz können trainiert werden, weitreichende und komplexe Aufgaben auszuführen. Jedoch handelt es sich dabei häufig um einen *Black-Box-Ansatz*, d.h. der Nutzer hat wenig bis keine Einsicht darin, wie Entscheidungen mithilfe dieses Modells getroffen werden. Besonders in sicherheitskritischen Anwendungen sollte man aber nicht blind auf gute Ergebnisse vertrauen, sondern in der Lage sein, gefällte Entscheidungen im Detail nachzuvollziehen. Wirklich wertvoll werden Modelle erst dann, wenn wir sie verstehen und erklären können.

Im Laufe dieser Arbeit werden wir feststellen, dass die klassische Hauptkomponentenanalyse ein solches Verständnis nicht ermöglicht. Häufig ist es unklar, was die neu kombinierten Variablen im Kontext repräsentieren. Die Möglichkeit einer Interpretation des Modells war von wesentlicher Bedeutung bei der Verwendung der dünnbesetzten Hauptkomponentenanalyse für den uns zur Verfügung stehenden Datensatz. Wir werden uns in Kapitel 6 genauer mit Frequenzdaten auseinandersetzen.

Aufbau der Arbeit

Eigene Beiträge

Ein Großteil der Arbeit zur dünnbesetzten Hauptkomponentenanalyse beruht auf dem von Zou und Haste in [ZHT06] eingeführten Ansatz. Er ist in der Literatur der wohl weitverbreiteste und hat den Grundstein für eine Verwendung der Methode in der Praxis gelegt. Wir möchten an dieser Stelle darauf hinweisen, dass während der Entstehung dieser Arbeit einige Kritikpunkte durch eine Veröffentlichung von Camacho et al. aufgezeigt wurden. Wir haben Änderungen wie Korrekturen eingearbeitet und werden Unterschiede im Folgendem deutlich machen. Insgesamt haben wir in dieser Arbeit folgende Eigenbeiträge geleistet.

- Detaillierte Aufarbeitung der mathematischen Grundlagen für die dünnbesetzte Hauptkomponentenanalyse
- Erläuterung der Konstruktion und Einarbeitung von theoretischen Resultaten
- Eigene, beschleunigte Implementierung des Algorithmus in Python
- Anwendung des Verfahrens auf Frequenzdaten

- Empirische Validierung der von Camacho et al. gewonnenen Kenntnisse
- Vergleich mit anderen Ansätzen sowie Ausblick für mögliche Verbesserungen

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 des linearen Algebra, Matrixzerlegungen und ausgewählten Approximationsproblemen in Abschnitt 2.1. Ein Großteil dieses Kapitels ist verallgemeinerten linearen Regressionsmodellen gewidmet. Ab Abschnitt 2.10 werden wir die Modelle zu Zwecken der Regularisierung mit Straftermen versehen, um verschiedene 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.

2.1 Lineare Algebra

Die Mathematik der Hauptkomponentenanalyse beruht im Wesentlichen 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 $\langle \cdot, \cdot \rangle : V \times V \longrightarrow \mathbb{R}$ mit den folgenden Eigenschaften:

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

$$\begin{aligned} \langle \cdot, x \rangle : V &\longrightarrow \mathbb{R} \\ v &\longmapsto \langle v, x \rangle \end{aligned} \qquad \begin{aligned} \langle x, \cdot \rangle : V &\longrightarrow \mathbb{R} \\ v &\longmapsto \langle x, v \rangle \end{aligned}$$

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 euklidischem 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 $(v_1, \ldots, v_n)$ in V heißt *orthogonal* oder *Orthogonalsystem*, wenn

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

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

Definition 2.4 (Orthogonale Matrix [Ant98]). Eine Matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ heißt *orthogonal*, falls deren Zeilen- und Spaltenvektoren paarweise orthonormal bezüglich des Standardskalarprodukts sind, d.h.

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

Definition 2.5 (Orthogonalprojektion [Ant98]). Eine *Orthogonalprojektion* auf einen Untervektorraum U eines Vektorraumes V ist eine lineare Abbildung $P_U \colon V \to V$, 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 Orthogonalbasis für U lässt sich aus dieser Definition eine explizite Lösung für die Orthogonalprojektion $P_U(v)$ herleiten.

Theorem 2.6 ([Ant98]). *Ist $(u_1, \ldots, u_r)$ eine Orthogonalbasis von U, so gilt für alle $v \in V$*

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

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

$$P_U(v) = \sum_{i=1}^{r} \frac{v^\top u_i}{u_i^\top u_i} u_i = \sum_{i=1}^{r} \frac{u_i u_i^\top}{u_i^\top u_i} v = \mathbf{A}\mathbf{A}^\top v, \tag{2.1}$$

wobei $\mathbf{A} = \begin{bmatrix} \frac{u_1}{\|u_1\|} & \cdots & \frac{u_r}{\|u_r\|} \end{bmatrix} \in \mathbb{R}^{n \times r}$. Die Orthogonalitätsbedingung in Theorem 2.6 kann auch weggelassen werden. Ist $(u_1, \ldots, u_r)$ eine beliebige Basis von U, so gilt

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

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

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

2.1.2 Matrixzerlegungen

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

Definition 2.8 (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 λ heißt *Eigenwert* von $\mathbf{A}$.

Definition 2.9 (Diagonalisierbarkeit [Ant98]). Eine quadratische Matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ heißt *diagonalisierbar*, falls 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 2.10 (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 $\lambda_1, \ldots, \lambda_r$ die verschiedenen Eigenwerte von $\mathbf{A}$.

Zusammenfassend besitzt eine symmetrische Matrix also immer 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 auch von nicht quadratischen oder nicht symmetrischen Matrizen.

Theorem 2.11 (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 2.12 (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 λ_i von $\mathbf{A}^T\mathbf{A}$. Konventionell werden die Singulärwerte von $\mathbf{D}$ absteigend sortiert, d.h. $\sigma_1 \geq \cdots \geq \sigma_r$. 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 2.13 (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 $\|\mathbf{A}\|_F^2 = \operatorname{tr}\left(\mathbf{A}^\top \mathbf{A}\right)$ gilt. Eine weitere wichtige Eigenschaft von Matrizen ist der *Rang*.

Definition 2.14 (Rang [Ant98])**.** Die Dimension des Zeilen- und des Spaltenraumes einer Matrix $\mathbf{A}$ heißt *Rang* von $\mathbf{A}$ und wird mit $\operatorname{rank}(\mathbf{A})$ bezeichnet.

Wir möchten nun eine Matrix $\mathbf{A}$ durch eine andere, simplere Matrix $\widehat{\mathbf{A}}$ mit niedrigerem Rang approximieren. Dieses Problem fällt unter die Kategorie *low rank approximation*, welche eine enge Verbindung zur Hauptkomponentenanalyse aufweist. Mithilfe der Singulärwertzerlegung können wir eine explizite Lösung angeben.

Theorem 2.15 (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:*

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

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)*

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

das Approximationsproblem

$$\min_{\operatorname{rank}(\widehat{\mathbf{A}})\leq r} \|\mathbf{A}-\widehat{\mathbf{A}}\|_F = \|\mathbf{A}-\widehat{\mathbf{A}}^*\|_F = \sqrt{\sigma_{r+1}^2+\cdots+\sigma_m^2}, \tag{2.3}$$

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

Das Eckart-Young-Mirsky-Theorem wird es uns in Abschnitt 3.4 ermöglichen, eine wertvolle Eigenschaft der Hauptkomponentenanalyse zu zeigen. In Anwendung korrespondieren die Singulärwerte mit dem Rekonstruktionsfehler und die Rang-Bedingung an $\widehat{\mathbf{A}}$ mit der Komplexität des Modells.

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 2.16 (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*

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

das Approximationsproblem

$$\widehat{\mathbf{A}} = \arg\min_{\mathbf{A}} \left\|\mathbf{M}-\mathbf{N}\mathbf{A}^\top\right\|_F^2 \quad \text{unter der Nebenbedingung, dass } \mathbf{A}^\top\mathbf{A} = \mathbb{1}_{k\times k}. \tag{2.4}$$

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

2.1.4 Pseudoinverse

In vielen Anwendungen der numerischen Mathematik benötigt man für die Angabe einer expliziten Lösung die Inverse einer Matrix. Allerdings sind diese nur für quadratische, nichtsinguläre Matrizen definiert. Daher ist eine Verallgemeinerung des Konzepts nötig. Verallgemeinerte Inversen werden in der Literatur nicht einheitlich gehandhabt und orientieren sich häufig an der zu lösenden Aufgabenstellung. Für unseren Anwendungsfall in Kapitel 4 benutzen wir die *Moore-Penrose-Inverse*. Wir werden die beiden Begriffe Pseudoinverse und Moore-Penrose-Inverse daher synonym verwenden.

Definition 2.17 (Moore-Penrose-Inverse [Isr03])**.** Die *Moore-Penrose-Inverse* einer Matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ ist die eindeutig bestimmte Matrix $\mathbf{A}^+ \in \mathbb{R}^{n\times m}$, welche die folgenden vier Eigenschaften erfüllt

(i) $\mathbf{A}\mathbf{A}^+\mathbf{A} = \mathbf{A}$

(ii) $\mathbf{A}^+\mathbf{A}\mathbf{A}^+ = \mathbf{A}^+$

(iii) $(\mathbf{A}\mathbf{A}^+)^T = \mathbf{A}\mathbf{A}^+$

(iv) $(\mathbf{A}^+\mathbf{A})^T = \mathbf{A}^+\mathbf{A}$

Für quadratische, nichtsinguläre Matrizen entspricht die Pseudoinverse der regulären Inversen $\mathbf{A}^+ = \mathbf{A}^{-1}$. Sind die Spalten bzw. Zeilen der Matrix $\mathbf{A}$ linear unabhängig, gilt $\mathbf{A}^+ = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T$ bzw. $\mathbf{A}^+ = \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}$. Im Allgemeinen kann die Pseudoinverse mithilfe einer Singulärwertzerlegung berechnet werden. Ist $\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^T$ eine Singulärwertzerlegung von $\mathbf{A}$, gilt $\mathbf{A}^+ = \mathbf{V}\mathbf{D}^+\mathbf{U}^T$. Für eine Diagonalmatrix $\mathbf{D}$ entsteht die Pseudoinverse durch Transponieren und Invertieren der von null verschiedenen Elemente

$$(\mathbf{D})^+_{ij} = \begin{cases} \frac{1}{\sigma_i} & \text{falls } i = j \wedge \sigma_i \neq 0 \\ 0 & \text{sonst} \end{cases} .$$

2.2 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. Für die Einführung generalisierter linearer Modelle richten wir uns vor allem nach [HTF09].

2.2.1 Norm

Definition 2.18 ([Hie19])**.** Eine Abbildung $\|\cdot\| : \mathbb{R}^n \longrightarrow \mathbb{R}_0^+$ 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 (ℓ_p-Norm [SW04])**.** Auf dem $\mathbb{R}^n$ sind die l_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 ℓ_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 ℓ_0-"Norm". Diese zählt die von null verschiedenen Einträge eines Vektors und misst, ob ein Vektor dünnbesetzt ist.

Definition 2.20 (ℓ_0-"Norm" [FR13])**.** Die sogenannte ℓ_0-"Norm" ist definiert durch

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

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

$$\|x\|_q^q = \sum_{i=1}^n |x_i|^q \quad \xrightarrow[q\to 0]{} \quad \sum_{i=1}^n \mathbf{1}_{\{x_i \neq 0\}} = |\{i\colon \ x_i \neq 0\,, \quad 1 \leq i \leq n\}|$$

Wir wollen betonen, dass die ℓ_0-"Norm" keine wirkliche Norm ist, da die Abbildung nicht homogen ist. Trotzdem ist diese "Norm" in der Theorie der komprimierten Erfassung 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.

2.2.2 Grundlagen aus der Statistik

Definition 2.21 (Stichprobenkovarianz [Rüs14])**.** Seien $(x_1, y_1), \dots (x_n, y_n)$ zweidimensionale Daten. Dann ist die *Stichprobenkovarianz* durch

$$s_{x,y} = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x}_n)(y_i - \bar{y}_n)$$

gegeben, wobei $\bar{x}_n = \frac{1}{n}\sum_{i=1}^n x_i$ das Strichprobenmittel bezeichnet.

Der Faktor $\frac{1}{n-1}$ wird als *Bessel-Korrektur* bezeichnet und stellt die Erwartungstreue des Schätzers sicher. Im Folgenden werden diesen Faktor allerdings ignorieren, da er für die von uns betrachteten Optimierungsprobleme in Kapitel 3 und 4 aufgrund der Normierung irrelevant ist. Für einen Datensatz $\mathbf{X} \in \mathbb{R}^{n\times p}$ mit p zentrierten Variablen werden wir häufig von der Stichprobenkovarianzmatrix $\mathbf{\Sigma}$ sprechen, welche nach obiger Definition bis auf die Bessel-Korrektur durch $\mathbf{X}^T\mathbf{X}$ gegeben ist.

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

Definition 2.22 (Verzerrung [Rüs14])**.** Die *Verzerrung* (Englisch: *Bias*) des Schätzers d bei ϑ ist definiert durch

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

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

Definition 2.23 (Mittlerer quadratischer Fehler [Koh05])**.** Der *mittlere quadratische Fehler* (Englisch: *Mean Squared Error (MSE)*) ist definiert durch

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

Theorem 2.24 (Verschiebungssatz [Koh05])**.** *Der mittlere quadratische Fehler zerfällt in Varianz und Bias, d.h.*

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

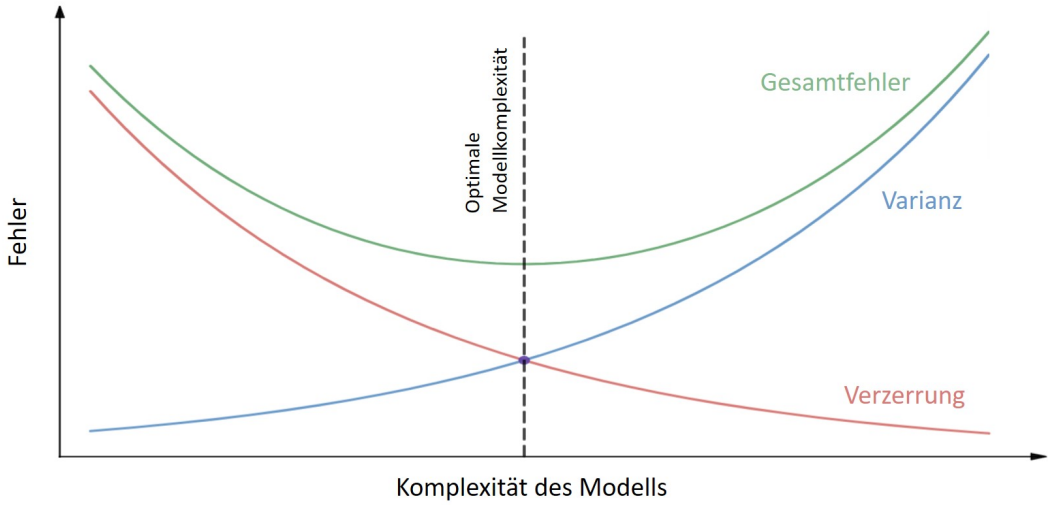


ABBILDUNG 2.1: Das Verzerrung-Varianz-Dilemma beschreibt den Kompromiss zwischen Unter- und Überanpassung an den Trainingsdatensatz, welchen man für ein robustes Modell eingehen muss.

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 jedoch 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 Abbildung 2.1 veranschaulicht.

2.2.3 Lineare Regression

Bei der Regressionsanalyse werden Zusammenhänge zwischen mehreren Merkmalen untersucht. Man versucht eine unabhängige Variable Y durch eine oder mehrere abhängige Variablen $X_1, \ldots, X_p$ 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 β_j die Regressionskoeffizienten sind. Bei der Verwendung dieses Modells nehmen wir an, dass die Regressionsfunktion $\mathrm{E}(Y|X)$ linear ist bzw. ein lineares Modell eine geeignete Approximation ist [HTF09].

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

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

und das dazugehörige Minimierungsproblem

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

wobei $\mathbf{X} \in \mathbb{R}^{n \times (p+1)}$ die Matrix der x_i mit einer 1 an erster Stelle ist und $y = (y_1, \ldots, y_n)^\top$. 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

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

besitzt. Die Zielgröße $\hat{y}$ ergibt sich dann durch

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

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 meist 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 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. Falls wir über mehr Variablen als Beobachtungen verfügen neigen lineare Regressionsmodelle 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 Null setzen. Des Weiteren erschwert eine hohe Anzahl an Variablen, die

in das Modell einfließen, 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.

2.2.4 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

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

wobei $\lambda_2 \geq 0$ ein Hyperparameter ist, der die Stärke der Schrumpfung der Regressionskoeffizienten kontrolliert. Hyperparameter werden zur Steuerung des Trainingsalgorithmus verwendet und müssen vorab festgelegt werden. Je größer λ_2, desto stärker ist die Schrumpfung der β_j. Durch die Einführung des ℓ_2-Strafterms garantiert (2.10) auch für $p > n$ eine eindeutige Lösung, da $\|\cdot\|_2^2$ streng konvex ist. Da β_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 $x_{ij} = x_{ij} - \bar{x}_j$. Die eindeutige Lösung der zentrierten Version von (2.10) ist dann durch

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

gegeben, wobei $\beta = (\beta_1, \ldots, \beta_p)$ 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 erst für $\lambda_2 \to \infty$ erreicht.

2.2.5 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 ℓ_1-Strafterms

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

Es wird also im Vergleich zu (2.10) lediglich die ℓ_2-Norm durch eine ℓ_1-Norm ausgetauscht. Bevor wir uns mit der Lösung dieses Problems beschäftigen, möchten wir erklären, warum die ℓ_1-Norm eine Dünnbesetzung der Koeffizienten hervorruft. Zunächst geben wir eine

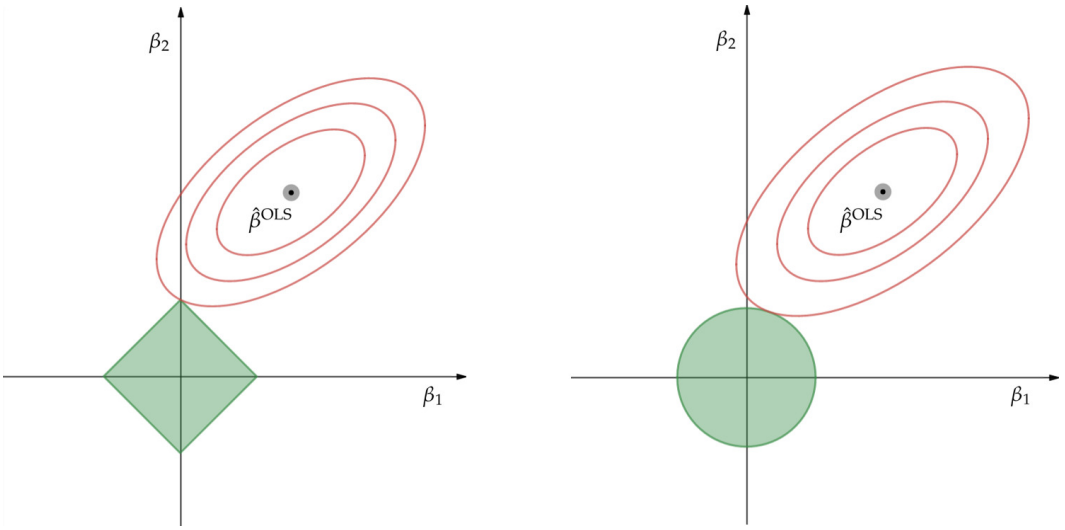


ABBILDUNG 2.2: Die Abbildung zeigt die Beschränkungen der ℓ_1-Norm (links) und der ℓ_2-Norm (rechts) zusammen mit den Höhenlinien der RSS-Funktion im $\mathbb{R}^2$. Verdeutlicht wird hier die geometrische Findung von $\hat{\beta}^{\text{lasso}}$ (links) und $\hat{\beta}^{\text{ridge}}$ (rechts). (Abbildung basiert auf [HTF09])

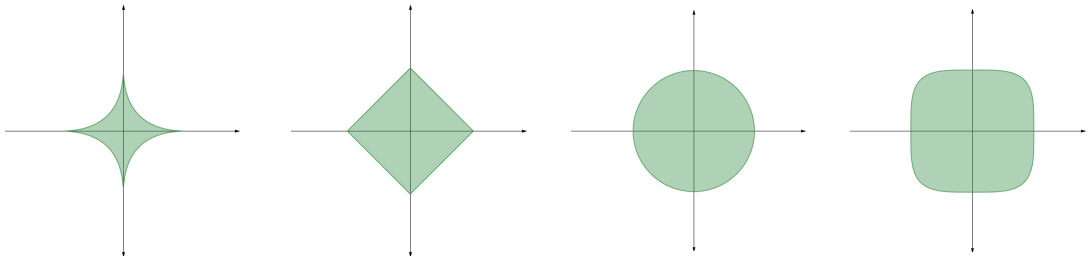
geometrische Erklärung, welche in Abbildung 2.2 zu sehen ist. Dort sind die ℓ_1- und ℓ_2-Beschränkungen, sowie die Höhenlinien der RSS-Funktion in zwei Dimensionen aufgezeichnet. Die optimalen Koeffizienten von Ridge und Lasso Regression ergeben sich nun aus dem Schnittpunkt der Höhenlinien mit der Begrenzung der jeweiligen Norm. Wählen wir die ℓ_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 gibt es bei einer ℓ_2-Begrenzung keine Ecken, weshalb jeder Randpunkt der Begrenzung als Schnittpunkt in Frage kommt. Dadurch 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 ℓ_q-Normen als Strafterm für Werte $q < 1$ in Betracht ziehen. In Abbildung 2.3 sind die Begrenzungen der ℓ_q-Normen für verschiedene Werte von q eingezeichnet. Für $q \to 0$ 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

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

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

Um eine mathematisch gründliche Erklärung für die Dünnbesetzung zu liefern, wenden



$q = 0.5$ $q = 1$ $q = 2$ $q = 4$

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

wir uns der Lösung von (2.12) zu. Falls die Spalten von $\mathbf{X}$ orthonormal sind, ergibt sich eine explizite Lösung

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

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

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

mithilfe eines Schwellwerts δ definiert wird. Dieser steht im Gegensatz zum hard-thresholding Operator und wird in Abbildung 2.4 dargestellt. Durch den soft-thresholding Operator werden den Koeffizienten in $\hat{\beta}^{\text{OLS}}$ entweder auf Null gesetzt oder um $\frac{\lambda_1}{2}$ geschrumpft. 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.

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+04b], 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 2.2.6 werden wir auf ein Koordinaten-Abstiegsverfahren weiter eingehen.

Es wurde herausgearbeitet, 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 Gruppen vorab im Datensatz festlegen kann. Der im Zuge dieser Arbeit aber wichtigere Nachteil ist, dass das Lasso maximal n Variablen selektieren kann, falls $p > n$ ist. In diesem Fall hat $\mathbf{X}$ maximal Rang n hat, weshalb wir y mithilfe von n Variablen perfekt vorhersagen können. 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 verschiedene Koeffizienten zuzulassen.

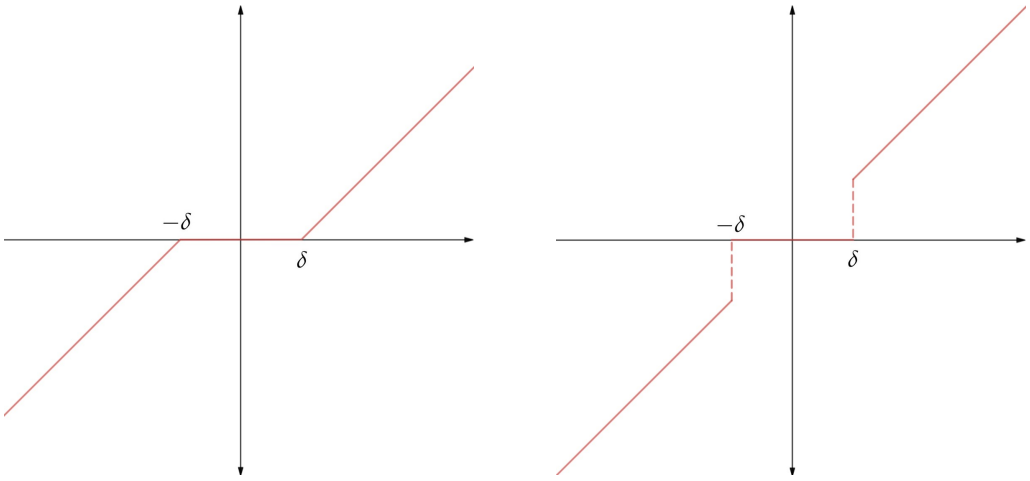


ABBILDUNG 2.4: Gezeigt werden zwei verschiedene Methoden für ein Schwellenwertverfahren. Bei der soft-thresholding-Operation $\text{soft}_\delta(x)$ (links) und hard-thresholding-Operation $\text{hard}_\delta(x)$ (rechts) werden alle Einträge von x kleiner als δ auf Null gesetzt. Der Unterschied besteht darin, dass $\text{soft}_\delta(x)$ die verbliebenen Einträge um $|\delta|$ schrumpft.

2.2.6 Elastic Net

Damit im Fall $p > n$ mehr als n Variablen für das Modell selektiert werden können, betrachten wir eine Kombination von Lasso und Ridge Regression. Durch die Einbettung einer ℓ_1 und ℓ_2-Norm erhalten wir das sog. *Elastic Net* [ZH05]

$$\hat{\beta}^{\text{en}} = \arg\min_{\beta} \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_2 \sum_{j=1}^{p} \beta_j^2. \tag{2.16}$$

Wie zuvor wird durch den ℓ_1-Strafterm ein dünnbesetztes Modell generiert. Dagegen fördert der ℓ_2-Strafterm den Gruppeneffekt, stabilisiert den Regularisierungspfad und erlaubt eine beliebige Anzahl zu selektierender Variablen. 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. Für eine allgemeine numerische Lösung wurden zum Beispiel LARS-EN [ZH05], welches auf dem LARS Algorithmus für das Lasso basiert, und ein Koordinaten-Abstiegsverfahren [FHT10] vorgeschlagen. Die Implementierung des Elastic Nets in scikit-learn [Ped+11] beruht auf letzterem Verfahren mit einer leicht abgeänderten mathematischen Formulierung

$$\arg\min_{\beta} R_\lambda(\beta) = \arg\min_{\beta} \frac{1}{2n} \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda\alpha \|\beta\|_1 + \frac{\lambda}{2}(1-\alpha)\|\beta\|_2^2, \tag{2.17}$$

wobei $\alpha \in [0,1]$ das Verhältnis zwischen ℓ_1- und ℓ_2-Norm ist. Während wir mit α das Verhältnis der beiden Regressionen bestimmen, können wir mit λ die Stärke der Bestrafung kontrollieren. Setzt man

$$\lambda = \frac{2\lambda_2 + \lambda_1}{2n} \quad \text{und} \quad \alpha = \frac{\lambda_1}{2\lambda_2 + \lambda_1} \tag{2.18}$$

entspricht (2.17) unserem ursprünglich formulierten Problem. Durch die Reparametrisierung können wir das Verhältnis der Bestrafungen flexibel anpassen. Mit $\alpha = 0$ reduziert sich (2.17) auf Ridge und für $\alpha = 1$ auf die Lasso Regression.

Oft ist es schwer, alle Koeffizienten gleichzeitig zu optimieren. Daher nutzt man bei einem Koordinaten-Abstiegsverfahren aus, dass eine partielle Optimierung über den j-ten Koeffizienten einfach zu berechnen ist, wenn alle anderen fixiert sind. Mithilfe der bisherigen Schätzer $\tilde{\beta}_0$ und $\tilde{\beta}_l$ für $l \neq j$, löst man also

$$\tilde{\beta}_j = \arg\min_{\beta_j} R_\lambda(\tilde{\beta}_0, \ldots, \tilde{\beta}_{j-1}, \beta_j, \tilde{\beta}_{j+1}, \ldots, \tilde{\beta}_p). \tag{2.19}$$

Da $R_\lambda(\tilde{\beta}_0, \ldots, \tilde{\beta}_{j-1}, \beta_j, \tilde{\beta}_{j+1}, \ldots, \tilde{\beta}_p)$ für $\beta_j = 0$ nicht differenzierbar ist, werden zur Lösung von (2.19) Subdifferentiale benutzt, welche eine Verallgemeinerung des Gradienten auf nicht differenzierbare konvexe Funktionen ist. Wir werden uns auf die Angabe einer Lösung beschränken und verweisen für eine Herleitung dieser auf [DJ94]. Der Übersichtlichkeit halber nehmen wir zusätzlich an, dass die Variablen standardisiert wurden, d.h. $\sum_{i=1}^n x_{ij} = 0$ und $\frac{1}{n-1}\sum_{i=1}^n x_{ij}^2 = 1$. Somit ist eine explizite Lösung von (2.19) durch

$$\tilde{\beta}_j = \frac{\text{soft}_{\lambda\alpha}(\frac{1}{n}\sum_{i=1}^n x_{ij}(y_i - (\tilde{\beta}_0 + \sum_{l\neq j} x_{il}\tilde{\beta}_l)))}{1 + \lambda(1-\alpha)} \tag{2.20}$$

gegeben. Dabei ist $y_i - (\tilde{\beta}_0 + \sum_{l\neq j} x_{il}\tilde{\beta}_l)$ das partielle Residuum der i-ten Beobachtung bei Anpassung von β_j, d.h. das Residuum bei bestmöglicher Vorhersage ohne Einbeziehung von β_j. Der Term $\frac{1}{n}\sum_{i=1}^n x_{ij}(y_i - (\tilde{\beta}_0 + \sum_{l\neq j} x_{il}\tilde{\beta}_l))$ ist ein Maß dafür, ob eine Einbeziehung von β_j in das Modell die Vorhersagen verbessert. Nach Berechnung dieses Werts, wenden wir soft-thresholding gemäß der ℓ_1-Bestrafung an und schrumpfen die Koeffizienten mit dem Faktor $1 + \lambda(1-\alpha)$ für den Beitrag der ℓ_2-Norm. Wegen der Standardisierung gilt

$$\frac{1}{n}\sum_{i=1}^n x_{ij}(y_i - (\tilde{\beta}_0 + \sum_{l\neq j} x_{il}\tilde{\beta}_l)) = \frac{1}{n}\sum_{i=1}^n x_{ij}r_i + \tilde{\beta}_j \tag{2.21}$$

wobei r_i das aktuelle Residuum der i-ten Beobachtung ist. Anhand von (2.21) können wir erkennen, warum das Koordinaten-Abstiegsverfahren effizient ist. Viele der Koeffizienten, die zuvor Null waren, bleiben nach der soft-thresholding Operation Null, wenn die Residuen bezüglich β_j nicht zu groß sind. Daher muss β_j in vielen Fällen nicht aktualisiert werden.

Für die Wahl des Koeffizienten, welcher als nächstes minimiert werden soll, gibt es verschiedene Möglichkeiten. Man kann zyklisch, zufällig oder in die Richtung des steilsten Abstiegs entlang der Koordinaten vorgehen. In Algorithmus 1 haben wir das zyklische Koordinaten-Abstiegsverfahren zusammengefasst.

Algorithm 1 Koordinaten-Abstiegsverfahren für das Elastic Net

procedure ELASTICNET($\mathbf{X}, y, \lambda, \alpha$)
  Initialisiere $\beta = (\mathbf{X}^T\mathbf{X} + \lambda\mathbb{1})^{-1}\mathbf{X}^\top y$
  while nicht konvergiert **do**
    for $j = 1, \ldots, p$ **do**
      $c_j \leftarrow \frac{1}{n}\sum_{i=1}^n x_{ij}(y_i - (\beta_0 + \sum_{l\neq j} x_{il}\beta_l))$
      $\beta_j \leftarrow \frac{\text{soft}_{\lambda\alpha}(c_j)}{1+\lambda(1-\alpha)}$

Auch wenn klassische Kriterien für die Konvergenz von Koordinaten-Abstiegsverfahren

	AGE	SEX	BMI	BP	...	Blutproben				...	Zielgröße
Patient	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+04a]. 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 (S1, ..., S6). Die Zielgröße y enthält Werte für den Krankheitsfortschritt ein Jahr nach Behandlungsbeginn.

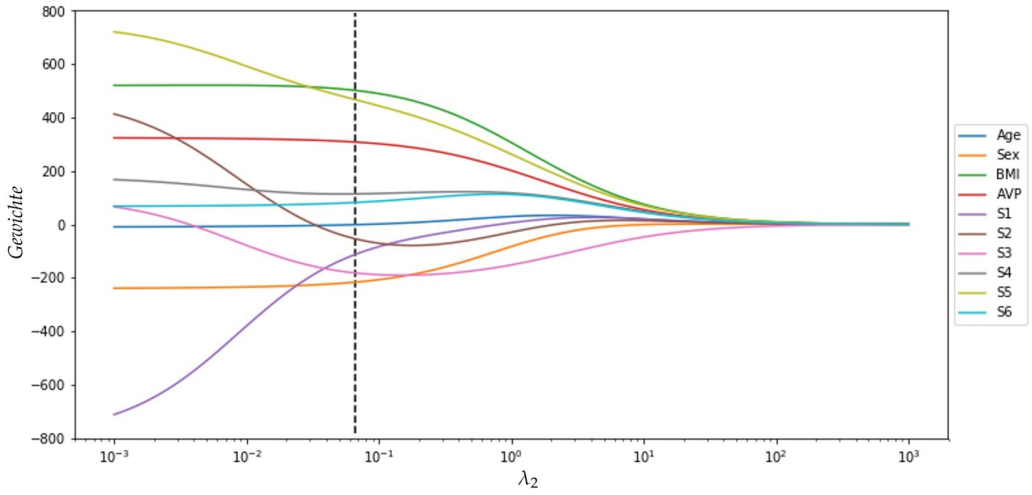
aufgrund der fehlenden Differenzierbarkeit nicht zutreffen, besitzt (2.17) eine andere charakterisierende Eigenschaft, für welche das Verfahren konvergiert [Tse+88].

2.2.7 Vergleich der Regressionsmethoden

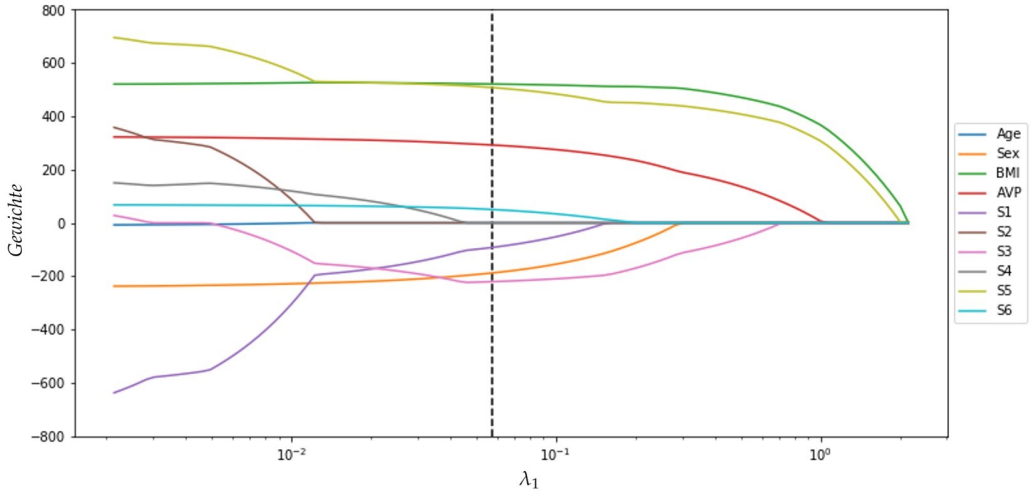
Zur Veranschaulichung der oben eingeführten Methoden werden wir diese nun anwenden. Dabei greifen wir auf einen durch scikit-learn [Ped+11] bereitgestellten Datensatz, der erstmals durch [Tib+04b] öffentlich gemacht worden ist, zurück. Ein Ausschnitt des Datensatzes befindet sich in Tabelle 2.1.

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 λ_1, λ_2 bzw. λ visuell nachvollziehen. Deutlich zu erkennen ist, dass bei der Ridge Regression eine kontinuierliche Schrumpfung der Koeffizienten stattfindet, d.h. sie werden erst für $\lambda_2 \to \infty$ auf Null gesetzt. Im Gegensatz dazu erkennt man bei der Lasso Regression, dass schon für kleine λ_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 werden und andererseits eine Stabilisierung der Regularisierungspfade.

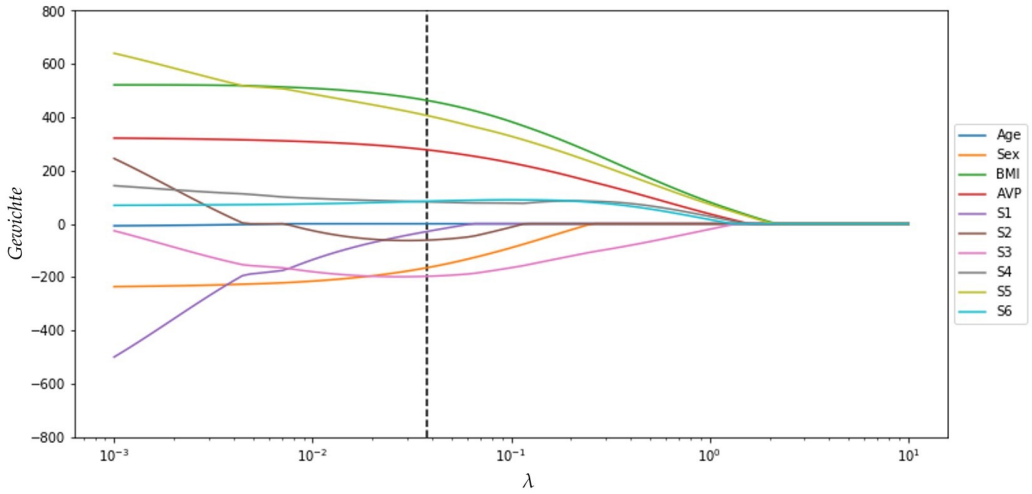
Um ein geeignetes Modell zu erhalten, müssen die Hyperparameter λ_1, λ_2 bzw. λ geeignet gewählt werden. Zu diesem Zweck kann ein k-faches Kreuzvalidierungsverfahren eingesetzt werden. Hierbei wird der Datensatz in k möglichst gleich große Teilmengen $T_1, \ldots, T_k$ zerlegt. Anschließend werden k Durchläufe gestartet, wobei die jeweils i-te Teilmenge T_i als Testmenge und die verbleibenden $k-1$ Teilmengen $\{T_1, ..., T_k\} \setminus \{T_i\}$ als Trainingsmenge verwendet werden. Mittelt man die mittleren quadratischen Fehler der Testmengen, erhält man ein Maß für dem Gesamtfehler. Man wählt dann den Regularisierungsparameter mit minimalem Gesamtfehler, welcher in Abbildung 2.5 eingezeichnet ist.



(A) Verhalten der Koeffizienten bei der Ridge Regression.



(B) Verhalten der Koeffizienten bei der Lasso Regression.



(C) Verhalten der Koeffizienten bei der Elastic Net Regression mit festem $\alpha = 0.99$.

ABBILDUNG 2.5: Jede Linie entspricht dem Wert (y-Achse) des jeweiligen Regressionskoeffizienten in Abhängigkeit der Regularisierungsparameter λ_2, λ_1 bzw. λ (x-Achse). Zu sehen ist die unterschiedliche Art der Schrumpfung für die verschiedenen Strafterme. Die vertikalen Linien entsprechen dem Wert des jeweiligen Hyperparameters, der durch ein 10-faches Kreuzvalidierungsverfahren bestimmt worden ist.

Kapitel 3

Hauptkomponentenanalyse

Die *Hauptkomponentenanalyse (Englisch: Principal Component Analysis (PCA))* ist ein weitverbreitetes multivariates statistisches Verfahren zur Dimensionsreduktion. Allgemein zielen derartige Verfahren darauf ab, die in einem Datensatz enthaltene Zahl an Variablen zu verringern, ohne dabei die darin enthaltene Information zu verlieren. Durch die 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.

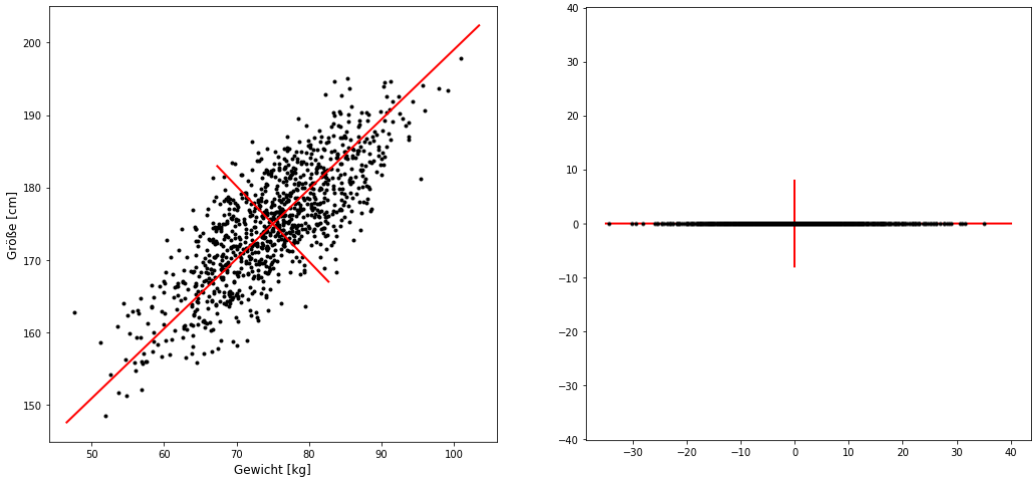
Aus diesem Grund hat die Hauptkomponentenanalyse in vielen Bereichen erfolgreich Anwendung gefunden. So kann es in der Bildverarbeitung beispielsweise zur Rauschunterdrückung [BSP12] oder zur Gesichtserkennung [Tai+17] genutzt werden. Um Bilder für solch ein Verfahren nutzbar zu machen, werden einzelne Pixel oder patches, also lokale Gruppierungen von Pixeln, als Variable interpretiert.

Mathematisch gesehen, kann die Hauptkomponentenanalyse auf verschiedene Weisen formuliert werden. Zunächst stellen wir die Idee des minimalen Informationsverlust in den Vordergrund. Anschließend werden wir die Verbindung zur Regressionsanalyse demonstrieren, welche als Grundlage für Kapitel 4 dient. Darüber hinaus wird die geometrische Interpretation des Verfahrens verdeutlicht und Grenzen aufgezeigt. Zum Schluss runden wir das Kapitel mit Verweisen auf 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. Dagegen sind Variablen mit niedrigere Varianz auf der Suche nach Unterschieden und Struktur im Datensatz nicht von Nutzen.

Um die Dimension zu reduzieren, könnte man einfach nach Eigenschaften größter Varianz suchen und alle Variablen unterhalb eines festgelegten Grenzwertes verwerfen. Dieser Art Vorgehen fallen 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



(A) Simulierte Daten zu Gewicht [kg] und Größe [cm] für 1000 Personen

(B) Daten nach der Transformation auf die erste Hauptachse

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.

neue Variablen, die sich aus den Bestehenden durch Linearkombinationen zusammensetzen. Variablen mit hoher Varianz werden in dieser 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 lässt sich ein Großteil an Information in nur 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 in diesem Beispiel noch immer knapp 90% der Varianz des ursprünglichen Datensatzes erklärt werden.

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. 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 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 Einheitsvarianz gebracht. Dies wird erreicht, indem man X_i durch $\frac{X_i - \mathrm{E}[X_i]}{\sqrt{\mathrm{Var}[X_i]}}$ ersetzt. Mathematisch gesehen wendet man das Verfahren somit auf die Stichprobenkorrelationsmatrix an.

3.1.1 Maximale Varianzerhaltung

Wir wollen nun die Intuition des minimalen Informationsverlusts 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 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 $\widehat{v}_1$, indem wir die Varianz entlang dieser maximieren, d.h.

$$\widehat{v}_1 = \underset{\|v\|_2=1}{\arg\max} \operatorname{Var}[\mathbf{X}v] = \underset{\|v\|_2=1}{\arg\max}\, v^\top \mathbf{\Sigma} v \tag{3.1}$$

wobei $\mathbf{\Sigma} = \mathbf{X}^\top \mathbf{X}$ die Stichprobenkovarianzmatrix bis auf die Bessel-Korrektur $\frac{1}{n-1}$ ist. Die restlichen Hauptachsen werden sequentiell definiert

$$\begin{gathered} \widehat{v}_{i+1} = \underset{\|v\|=1}{\arg\max}\, v^\top \mathbf{\Sigma} v \\ \text{unter der Nebenbedingung, dass } \widehat{v}_{i+1}^\top \widehat{v}_l = 0 \quad \forall 1 \leq l \leq i. \end{gathered} \tag{3.2}$$

Man sucht also unter den Richtungen, die orthogonal zu allen bisherigen Hauptachsen sind, diejenige, die die Varianz maximiert. Durch Transformation der Daten $\widehat{Z}_i = \mathbf{X}\widehat{v}_i$ 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 Langrange-Form $L(v, \lambda) = v^\top \mathbf{\Sigma} v + \lambda(1 - v^\top v)$ nach v ab, erhalten wir die notwendige Bedingung

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

und somit den stationären Punkt $\mathbf{\Sigma}\widehat{v}_1 = \lambda_1 \widehat{v}_1$. Das bedeutet, dass die erste Hauptachse genau dem Eigenvektor des größten Eigenwertes λ_1 der Stichprobenkovarianzmatrix $\mathbf{\Sigma}$ entspricht. Durch Linksmultiplikation mit $\widehat{v}_1^\top$ sehen wir, dass durch $\widehat{v}_1^\top \mathbf{\Sigma} \widehat{v}_1 = \lambda_1$ die Varianz der ersten Hauptkomponente gegeben ist. Analog zeigt man, dass auch die folgenden Hauptachsen, die durch (3.2) definiert sind, den Eigenvektoren von $\mathbf{\Sigma}$ entsprechen [Bis06].

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

$$\mathbf{\Sigma} = \widehat{\mathbf{V}} \widehat{\mathbf{\Lambda}} \widehat{\mathbf{V}}^\top,$$

wobei $\widehat{\mathbf{\Lambda}}$ eine Diagonalmatrix mit Eigenwerten λ_i und $\widehat{\mathbf{V}}$ die Matrix der Eigenvektoren ist. Somit können die Hauptachsen direkt aus $\widehat{\mathbf{V}}$ abgelesen werden. Die Hauptkomponenten

werden dann wie zuvor durch Multiplikation der Beobachtungen mit den Eigenvektoren erreicht.

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

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

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

$$\mathbf{X} = \widehat{\mathbf{U}}\widehat{\mathbf{D}}\widehat{\mathbf{V}}^\top$$

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

$$\mathbf{\Sigma} = \mathbf{X}^\top\mathbf{X} = \widehat{\mathbf{V}}\widehat{\mathbf{D}}\widehat{\mathbf{U}}^\top\widehat{\mathbf{U}}\widehat{\mathbf{D}}\widehat{\mathbf{V}}^\top = \widehat{\mathbf{V}}\widehat{\mathbf{D}}^2\widehat{\mathbf{V}}^\top.$$

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

3.1.2 Verbindung zur Regressionsanalyse

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. Mathematisch ausgedrückt minimieren wir also die Residuen der Projektion auf den Unterraum.

Sei dazu x_i die i-te Beobachtung und $\mathbf{V}_k = \begin{bmatrix} v_1 & \cdots & v_k \end{bmatrix}$ eine $p \times k$ orthonormale Matrix. Wie in Abschnitt 2.1.1 beschrieben, wird durch den Operator $\mathbf{V}_k\mathbf{V}_k^\top$ jede Beobachtung orthogonal auf den durch $v_1, \ldots, v_k$ aufgespannten Unterraum projiziert. Eine bestmögliche ℓ_2-Approximation der Daten ist gegeben, wenn wir die Distanz zwischen jeder Beobachtung und seiner Projektion minimieren [ZHT06]:

$$\widehat{\mathbf{V}}_k = \arg\min_{\mathbf{V}_k} \sum_{i=1}^{n} \left\| x_i - \mathbf{V}_k\mathbf{V}_k^\top x_i \right\|_2^2 \qquad (3.3)$$
$$\text{unter der Nebenbedingung, dass } \mathbf{V}_k^\top\mathbf{V}_k = \mathbb{1}_{k\times k}$$

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. Sprechen wir von der Varianz eines Datensatzes, meinen wir die Summe der Varianzen der einzelnen Variablen. Somit ist die Gesamtvarianz durch $\|\mathbf{X}\|_F^2$ gegeben. Aufgrund der orthogonalen Projektion erhalten wir mithilfe des verallgemeinerten

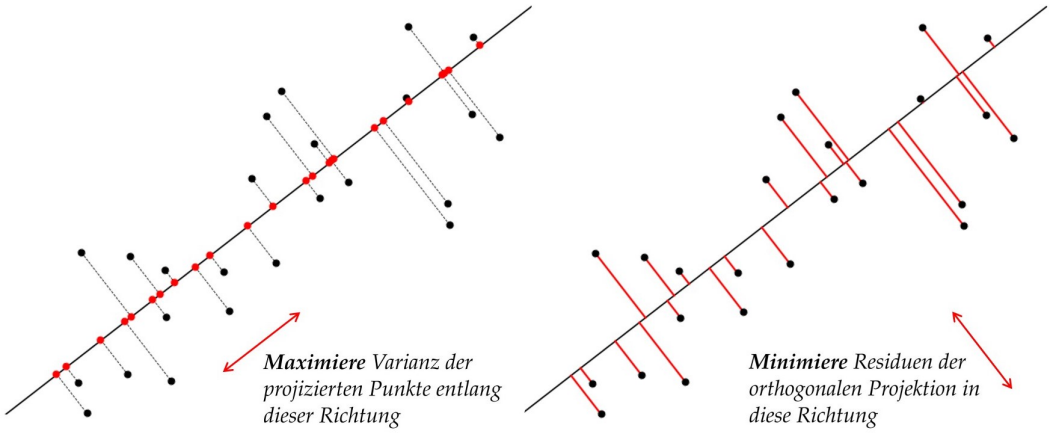


ABBILDUNG 3.2: Die Abbildung zeigt die Äquivalenz von Maximierung der Varianz und Minimierung der Residuen in zwei Dimensionen.

Satzes von Pythagoras

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

Weiter sehen wir, dass

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

und somit

$$\|\mathbf{X}\|_F^2 = \left\| \mathbf{X} - \mathbf{X}\mathbf{V}_k\mathbf{V}_k^\top \right\|_F^2 + \sum_{i=1}^{k} v_i^\top (\mathbf{X}^\top \mathbf{X}) v_i .$$

Nun sind wir in der Lage, die Verbindung zur ursprünglichen Konstruktion in (3.1) und (3.2) zu sehen. Dort haben wir den Term $v_i^\top (\mathbf{X}^\top \mathbf{X}) v_i$, der die Varianz der i-ten Hauptkomponente beschreibt, sequentiell maximiert. Mathematisch gesehen macht es keinen Unterschied, ob wir die Varianz der Hauptkomponenten $\sum_{i=1}^{k} v_i^\top (\mathbf{X}^\top \mathbf{X}) v_i$ maximieren oder die Residuen der Projektion $\left\| \mathbf{X} - \mathbf{X}\mathbf{V}_k\mathbf{V}_k^\top \right\|_F^2$ minimieren. Diese Idee ist in Abbildung 3.2 geometrisch veranschaulicht. Im linken Bild versucht man eine Richtung durch den Datensatz zu finden, welche die Varianz entlang dieser maximiert. Im Gegensatz dazu sucht man im rechtem Bild nach einer Richtung, welche die Summe der Distanzen zwischen Datenpunkt und seiner Projektion minimiert. In beiden Fällen resultiert dieselbe Hauptachse.

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 abweichenden Motivation der beiden Verfahren liegt der entscheidende Unterschied 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 in Kapitel 4 die Variante der dünnbesetzten Hauptkomponentenanalyse beschreiben.

3.1.3 Weitere Formulierungen

Wir werden nun kurz auf zwei weitere Formulierungen eingehen, anhand welcher weitere Eigenschaften der Hauptkomponentenanalyse deutlich werden. Ersetzt man in (3.3) $\mathbf{X}\widehat{\mathbf{V}}_k^\top$ durch die Hauptkomponenten $\widehat{\mathbf{Z}}_k$ erhält man

$$\begin{gathered}(\widehat{\mathbf{Z}}_k, \widehat{\mathbf{V}}_k) = \underset{\mathbf{Z}_k, \mathbf{V}_k}{\arg\min} \left\| \mathbf{X} - \mathbf{Z}_k \mathbf{V}_k^\top \right\|_F^2 \\ \text{unter der Nebenbedingung, dass } \mathbf{V}_k^\top \mathbf{V}_k = \mathbb{1}_{k\times k}.\end{gathered} \tag{3.4}$$

Durch $\widehat{\mathbf{Z}}_k \widehat{\mathbf{V}}_k^\top$ ist eine bestmögliche Rekonstruktion der Datenmatrix $\mathbf{X}$ gegeben. In anderen Worten wird in (3.4) also der ℓ_2-Rekonstruktionsfehler minimiert. Diese Formulierung wird häufig für Verallgemeinerungen der Hauptkomponentenanalyse gewählt und werden wir in Abschnitt 4.4 gebrauchen.

Für eine letzte Formulierung erinnern wir uns, dass die abgeschnittene Singulärwertzerlegung $\widehat{\mathbf{X}}_k = \widehat{\mathbf{U}}_k \widehat{\mathbf{D}}_k \widehat{\mathbf{V}}_k^\top \in \mathbb{R}^{n\times k}$ die optimale Lösung für die Hauptkomponentenanalyse liefert. Diese ist aufgrund des Eckart-Young-Mirsky-Theorem aus Abschnitt 2.1.3 ebenfalls Lösung des Problems

$$\begin{gathered}\widehat{\mathbf{X}}_k = \underset{\mathbf{X}_k}{\arg\min} \|\mathbf{X} - \mathbf{X}_k\|_F^2 \\ \text{unter der Nebenbedingung, dass } \operatorname{rank}(\mathbf{X}_k) \leq k.\end{gathered} \tag{3.5}$$

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

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

Infolgedessen können wir Bewertungskriterien definieren, welche es uns ermöglichen, verschiedene Modelle miteinander zu vergleichen. Wir wollen uns nun mit der Frage beschäftigen, wie stark wir die Dimension eines Datensatzes reduzieren können.

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 k bzw. der Rang von $\widehat{\mathbf{X}}_k$. Arbeiten wir auf rauschfreien Daten können wir k durch $\operatorname{rank}(\mathbf{X})$ schätzen. In der Regel ist unser Datensatz aber durch Rauschen gestört, weshalb $\mathbf{X}$ vollen Rang hat.

Aufgrund der Effizienz der Singulärwertzerlegung 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 σ_i^2 der einzelnen Hauptkomponenten. So können wir k zum Beispiel so wählen, dass der Rekonstruktionsfehler durch einen Parameter τ beschränkt ist

$$\widehat{k} = \min_k \left\{ k \colon \sum_{i=k+1}^{p} \sigma_i^2 < \tau \right\} \quad \text{oder} \quad \widehat{k} = \min_k \left\{ k \colon \sigma_{k+1}^2 < \tau \right\}. \tag{3.6}$$

Das zweite Auswahlkriterium wird in Abbildung 3.3 veranschaulicht. In der Praxis ist es aber sehr schwer, τ angemessen zu wählen, da die Singulärwerte von $\mathbf{X}$ nicht invariant unter Skalierung sind. Daher werden die Singulärwerte meist normiert

$$\widehat{k} = \min_k \left\{ k \colon \frac{\sum_{i=k+1}^{p} \sigma_i^2}{\sum_{i=1}^{k} \sigma_i^2} < \tau \right\} \quad \text{oder} \quad \widehat{k} = \min_k \left\{ k \colon \frac{\sigma_{k+1}^2}{\sum_{i=1}^{k} \sigma_i^2} < \tau \right\}. \tag{3.7}$$

Ersteres Auswahlkriterium in (3.7) ist weit verbreitet. Man wählt genau so viele Hauptkomponenten aus, dass höchstens ein gewisser Anteil der Varianz verloren geht. Typische Werte für τ liegen zwischen 10% und 20%.

In manchen Fällen gibt es eine klare Trennung in der Größe der Singulärwerte. Vidal et al. [VMS16] definieren ein Kriterium

$$\widehat{k} = \arg\min_k \quad \alpha\sigma_{k+1}^2 + \beta k, \tag{3.8}$$

welches im Wesentlichen nach einem starken Einbruch der Singulärwerte sucht. Mit $\alpha, \beta > 0$ haben wir damit direkten Einfluss auf das Verhältnis zwischen Rekonstruktionsfehler und Modellkomplexität. In Abbildung 3.3 haben wir die resultierende Gerade eingezeichnet.

In der Literatur existieren weitere Heuristiken, auf die wir hier nicht näher eingehen werden. Bis vor Kurzem gab es keinerlei theoretische Resultate für die 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

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

für $n \times n$-Matrizen minimal wird, falls alle Singulärwerte kleiner als $\frac{4}{\sqrt{3}}\sqrt{n}\sigma \approx 2.309\omega$ auf Null gesetzt werden. Wenn die Stärke des Rauschens ω vorher nicht bekannt ist, ändert sich die Schranke zu $2.858\omega_{med}$, wobei ω_{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.

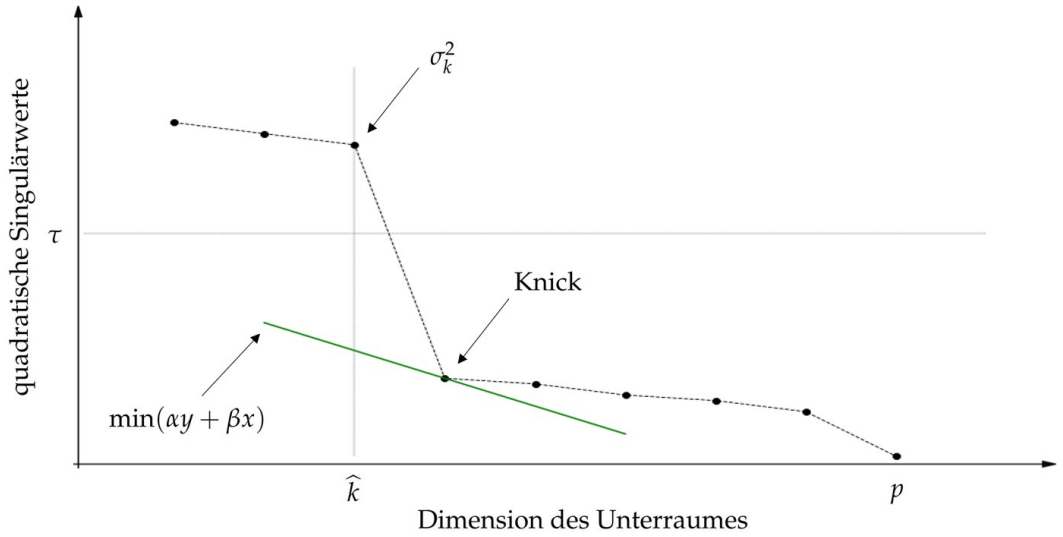


ABBILDUNG 3.3: Die Abbildung zeigt einen sog. *Scree Plot*, welche die quadratischen Singulärwerte bzw. die erklärte Varianz jeder einzelnen Hauptkomponente zeigt. Die Darstellung wird häufig genutzt, um zu entscheiden wie viele Hauptkomponenten für ein geeignetes Modell auszuwählen sind. (Abbildung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

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 Folgendem 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:

- Existenz linearer Beziehungen 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. *Kernel PCA* wendet die Hauptkomponentenanalyse in diesem Raum an bevor die Daten wieder zurück transformiert werden. 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. Falls dies nur einen Bruchteil der Daten betrifft, können durch entsprechende Erweiterungen mit hoher Wahrscheinlichkeit noch immer exakte Ergebnisse erzielt werden [Can+11]. 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 vorab unterschiedlich zu gewichten basierend auf deren geschätzten Relevanz. 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. In Theorem 3.2 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 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)*, welchem das folgende Kapitel 4 gewidmet ist.

3.4 Theoretische Aussagen

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

Eine wichtige Eigenschaft ist die Unkorreliertheit der Hauptkomponenten. Dies lässt sich anhand der Stichprobenkovarianzmatrix der Hauptkomponenten $\widehat{\mathbf{Z}}^\top\widehat{\mathbf{Z}} = (\mathbf{X}\widehat{\mathbf{V}})^\top\mathbf{X}\widehat{\mathbf{V}} = \widehat{\mathbf{V}}^\top\boldsymbol{\Sigma}\widehat{\mathbf{V}}$ sehen, welche aufgrund von $\widehat{\mathbf{V}}^\top\widehat{\mathbf{V}} = \mathbb{1}_{k\times k}$ eine Diagonalmatrix ist. Somit können wir von einer Hauptkomponente und deren erklärte Varianz sprechen, ohne uns dabei auf andere beziehen zu müssen.

Für unseren Anwendungsfall in Kapitel 6 befinden wir uns in einem $p \gg n$ Szenario. 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 $n \to \infty$ mit fixierter Dimension p.
- In der Theorie der Zufallsmatrizen erhöhen wir sowohl n als auch p mit der Voraussetzung, dass das Verhältnis $\frac{p_n}{n}$ asymptotisch konstant ist.
- In einem *high dimension low sample size setting (HDLSS)* betrachten wir die Asymptotik für $p \to \infty$ 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 $\mathbf{\Sigma}$ 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 $\widehat{v}_i$ ist mit der Populationshauptachse v_i

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

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

Abhängig von n, p und der Stärke des Rauschens im Modell ergeben sich unterschiedliche Ergebnisse für die Konsistenz der Hauptkomponentenanalyse. Allgemein ist die Tendenz, dass eine hohe Anzahl an Beobachtungen die Konsistenz der Hauptachsen fördert, während eine erhöhte Dimension eine Inkonsistenz hervorruft. Nicht überraschend kann man zeigen, dass die Eigenvektoren der Stichprobenkovarianzmatrix $\widehat{v}_i$ konsistente Schätzer für die Eigenvektoren der Kovarianzmatrix v_i sind, falls wir die Dimension fixieren und die Anzahl an Beobachtungen erhöhen $n \to \infty$ [And03]. Für unser Szenario jedoch interessanter sind hochdimensionale Fälle. Weil eine Betrachtung einer HDLSS-Situation die Einführung zu vieler neuer Prinzipien erfordern würde, präsentieren wir zum Abschluss dieses Kapitels ein Inkonsistenz-Theorem für den Fall eines asymptotisch konstanten Verhältnisses zwischen n und p. Dafür betrachten wir ein *single component model* [JY09]

$$x_i = l_i v_1 + \sigma r_i \quad i = 1, \dots n, \tag{3.9}$$

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 $r_i \sim N_p(0, \mathbb{1})$ unabhängige Rauschvektoren. Wegen der Rauscheffekte hat die Stichprobenkovarianzmatrix $\mathbf{\Sigma}$ fast sicher $\min\{n, p\}$ von null verschiedene Eigenvektoren. Sei $\widehat{v}_1$ der größte Stichprobeneigenvektor von $\mathbf{\Sigma}$. 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 σ die Stärke des Rauschens im Modell (3.9) und $\lim_{n\to\infty} \frac{\|v_1\|^2}{\sigma^2} = \omega > 0$ das limitierende Signal-Rausch-Verhältnis. Weiterhin sei*

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

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

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

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

Wir sehen, dass $R_\infty^2(\omega, c) < 1$ genau dann, wenn $c > 0$. Daher ist $\widehat{v}_1$ genau dann ein konsistenter Schätzer für v_1, wenn $c = 0$. Für $c > 0$ ist die Hauptkomponentenanalyse somit 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. In diesem Fall kann Konsistenz nur dann gewährleistet werden, wenn es eine deutliche Trennung in der Größe der Eigenwerte gibt. Für Details weiterer spezieller Bedingungen 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. Daher sind wir oft bereit, einen Teil der Varianz für eine vereinfachte Interpretation einzutauschen.

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 dem von Zou und Hastie in [ZHT06] eingeführten Ansatz intensiv beschäftigen. Dieser gilt sicherlich zu den meistverbreitesten Varianten der dünnbesetzten Hauptkomponentenanalyse. In einem erst vor Kurzem erschienenen wissenschaftlichen Artikel werden einige Probleme des Ansatzes deutlich. Darum erläutern wir in Abschnitt 4.4 Neuerungen und Korrekturen der Methode durch Camacho et al. [Cam+20]. Zum Schluss dieses Kapitels werden wir Möglichkeiten für eine automatisierte Wahl der Modellparameter präsentieren und die Konsistenz in hochdimensionalen Fällen darlegen.

4.1 Problemformulierung

Wir möchten nun Hauptachsen eines gegebenen Datensatzes identifizieren mit der Zusatzbedingung, dass diese dünnbesetzt sind. Die wohl einfachste Vorgehensweise ist, eine Schwellenwertmethode auf die durch die klassische Hauptkomponentenanalyse entstandenen 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 Koeffizienten bestimmt. Zu berücksichtigen sind unter anderem sowohl die Standardabweichung als auch die Korrelationen mit anderen Variablen. Bei einer Schwellenwertmethode werden diese Faktoren nicht beachtet, weshalb den Ergebnissen im Allgemeinen nicht vertraut werden darf.

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:

$$\begin{aligned} v_k &= \underset{\|v\|_2=1}{\arg\max}\, v^\top \mathbf{\Sigma} v \\ \text{wobei } v_k^\top v_l &= 0 \quad \forall 1 \leq l < k \quad \text{für } k \geq 2 \\ &\text{und } \|v_k\|_0 \leq t \end{aligned} \tag{4.1}$$

wobei $\mathbf{\Sigma} = \mathbf{X}^\top \mathbf{X}$ die Stichprobenkovarianzmatrix ist. Der einzige Unterschied zur klassischen Hauptkomponentenanalyse in (3.1) und (3.2) besteht in der Einführung der ℓ_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) und (3.2). 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.

SCoTLASS

Inspiriert von der Lasso Regression [Tib96] schlugen Jolliffe et al. [JTU03] vor, die ℓ_1-Norm anstelle der ℓ_0-Norm als Strafterm zu verwenden. Wie haben bereits in Abschnitt 2.2.5 beobachten können, dass die ℓ_1-Norm genutzt werden kann, um dünnbesetzte Vektoren zu erhalten. Somit liegt es nahe das Problem wie folgt zu formulieren.

$$\begin{aligned} v_k &= \underset{\|v\|_2=1}{\arg\max}\, v^\top \mathbf{\Sigma} v \\ \text{wobei } v_k^\top v_l &= 0 \quad \forall 1 \leq l < k \quad \text{für } k \geq 2 \\ &\text{und } \|v_k\|_1 \leq t \end{aligned} \tag{4.2}$$

Wie in (4.1) hat man mit dem Hyperparameter 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.

Weitere Relaxationsideen

Weitere Ideen zur Relaxation von (4.1) beruhen auf den unterschiedlichen Formulierungen der Hauptkomponentenanalyse. Durch die Einbettung von Straftermen sind die verschiedenen Sichtweisen bei der dünnbesetzten Variante aber nicht mehr äquivalent. Daher haben

wir eine selektive Übersicht der verschiedenen Ansätze erstellt. Ein interessierter Leser sei auf die folgenden Quellen verwiesen.

- eine iterative Schwellenwertmethode [SH08; WTH09]
- eine verallgemeinerte Potenzmethode [Jou+10]
- ein alternierendes Maximierungs-Netzwerk [Ric12]
- Vorwärts- und Rückwärts-Greedy-Suche mittels Branch-and-Bound-Verfahren [MWA06]
- Konvexe Relaxation mittels semidefiniter Programmierung [dAs+07]
- eine Bayes-Formulierung [GD09]

4.3 Konstruktion

Wir werden uns nun mit dem von Zou, Hastie und Tibshirani in [ZHT06] eingeführten Ansatz ausführlich beschäftigen. Ausgangspunkt ist die Verbindung zur Regression, welche wir in (3.3) beschrieben haben. Im Folgenden bezeichnet k die Anzahl an Hauptkomponenten, die wir extrahieren möchten und x_i die i-te Zeile von $\mathbf{X}$. Mit $\widehat{\mathbf{B}}$ werden wir die dünnbesetzten Hauptachsen bezeichnen, um sie von den klassischen Hauptachsen $\widehat{\mathbf{V}}$ zu unterscheiden. Das folgende Theorem erweitert die bisherige Formulierung, indem nicht nur ausschließlich orthogonale Projektionen erlaubt werden.

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

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

unter der Nebenbedingung, dass $\mathbf{A}^\top\mathbf{A} = \mathbb{1}_{k\times k}$.

Dann ist $\widehat{\beta}_j \propto \widehat{v}_j$ *für* $j = 1, 2, \ldots, k$.

Fordern wir $\mathbf{A} = \mathbf{B}$ reduziert sich die Verlustfunktion $\sum_{i=1}^{n} \left\| x_i - \mathbf{A}\mathbf{B}^\top x_i \right\|^2$ auf die klassische Hauptkomponentenanalyse in (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 ℓ_1-Strafterm in die Zielfunktion einbetten. Dafür definieren wir das *Sparse PCA Kriterium* mit den Hyperparametern $\lambda_{1,j}$ und λ_2

$$(\widehat{\mathbf{A}}, \widehat{\mathbf{B}}) = \arg\min_{\mathbf{A},\mathbf{B}} \sum_{i=1}^{n} \left\| x_i - \mathbf{A}\mathbf{B}^\top x_i \right\|_2^2 + \lambda_2 \sum_{j=1}^{k} \|\beta_j\|_2^2 + \sum_{j=1}^{k} \lambda_{1,j} \|\beta_j\|_1 \qquad (4.3)$$
$$\text{unter der Nebenbedingung, dass } \mathbf{A}^\top\mathbf{A} = \mathbb{1}_{k\times k}.$$

Die normierten Spalten von $\mathbf{B}$ nennen wir die *dünnbesetzten Hauptachsen*. Um die Dünnbesetzung für jede Hauptachse unterschiedlich wählen zu können, erlauben wir unterschiedliche Bestrafungen $\lambda_{1,j}$. Im Gegensatz dazu lassen wir für den Ridge-Strafterm keine differenzierte Behandlung zu, der für die Reduktion von (4.3) auf (3.3) benötigt wird falls $\lambda_{1,j} = 0$. Allerdings hat die ℓ_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 durch (4.3) im Gegensatz zu manch anderen Varianten der dünnbesetzten Hauptkomponentenanalyse eine zeitgleiche anstatt einer sequentiellen Berechnung der Hauptachsen erfolgt. 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 (4.3) nicht, so dass durchaus starke Korrelation zwischen den Hauptkomponenten auftreten kann. Während dies eine flexiblere Modellierung ermöglicht, wird eine geeignete Visualisierung schwieriger. 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 der erklärten Varianz ist. Oft kann ein Großteil an Information eines Datensatzes durch eine geringe Anzahl an Hauptkomponenten beschrieben werden, welches die Komplexität hochdimensionaler Daten verringert. 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.

Camacho et al. zeigen, dass viele der Varianten der dünnbesetzten Hauptkomponentenanalyse bezüglich dieser Aspekte 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 die Hauptkomponenten $\widehat{\mathbf{Z}}$ bislang wie bei der klassischen Hauptkomponentenanalyse berechnet, indem man $\widehat{\mathbf{Z}} = \mathbf{X}\widehat{\mathbf{B}}$ setzt [ZHT06]. Allerdings vernachlässigt man in diesem Fall, dass die Hauptachsen nicht orthogonal zueinander sind. Dies wird deutlich, wenn wir die Hauptkomponentenanalyse wie in (3.4) als eine bestmögliche Rekonstruktion der Datenmatrix $\mathbf{X}$ auffassen

$$\mathbf{X} = \widehat{\mathbf{Z}}\widehat{\mathbf{B}}^\top + \widehat{\mathbf{E}}, \tag{4.4}$$

wobei $\widehat{\mathbf{E}}$ die Matrix der Residuen ist. Ist uns eine volle Rang Approximation $\mathbf{X} = \widehat{\mathbf{Z}}\widehat{\mathbf{B}}^\top$ gegeben, können wir beide Seiten mit $\widehat{\mathbf{B}}$ multiplizieren, um die Hauptkomponenten zu erhalten

$\mathbf{X}\widehat{\mathbf{B}} = \widehat{\mathbf{Z}}\widehat{\mathbf{B}}^\top\widehat{\mathbf{B}} = \widehat{\mathbf{Z}}$. Letzterer Schritt ist aber nur gültig, falls die Hauptachsen orthogonal zueinander sind. Daher muss bei der dünnbesetzten Variante mit der Moore-Penrose-Inversen $(\widehat{\mathbf{B}}^\top\widehat{\mathbf{B}})^+$ korrigiert werden. Demnach sollten die Hauptkomponenten durch

$$\widehat{\mathbf{Z}} = \mathbf{X}\widehat{\mathbf{B}}^\top(\widehat{\mathbf{B}}^\top\widehat{\mathbf{B}})^+ \tag{4.5}$$

berechnet werden. Wir möchten an dieser Stelle anmerken, dass durch (4.5) keine sequentielle Berechnung der Hauptkomponenten mehr möglich ist. Es können also nicht ohne weiteres mehr Hauptkomponenten zum Modell hinzugefügt werden, da jede Hauptkomponente von allen Hauptachsen abhängt und somit eine Neuberechnung erfordert. Im Gegenzug zeigen Camacho et al. empirisch, dass die Korrelation zwischen den Hauptkomponenten durch (4.5) deutlich sinkt, was für viele Anwendungen von Vorteil sein kann.

Für die Modellbewertung wird oft der Anteil erklärter 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 folgende Methode vor. Die erklärte Varianz für die ersten $j+1$ Hauptkomponenten sollte sich aus der Summe der ersten j und der erklärten Varianz der $j+1$-ten Hauptkomponente $\widehat{Z}_{j+1}$ ergeben. Aufgrund der Korrelation erhält die Varianz von $\widehat{Z}_{j+1}$ aber Beiträge anderer Hauptkomponenten. Um nur die zusätzlich durch $\widehat{Z}_{j+1}$ erhaltene Varianz zu erhalten und lineare Abhängigkeiten zu entfernen, nutzen Zou et al. eine Projektion

$$\widehat{Z}_{j\cdot 1,\dots,j-1} = \widehat{Z}_j - \mathbf{P}_{1,\dots,j-1}\widehat{Z}_j \tag{4.6}$$

wobei $\mathbf{P}_{1,\dots,j-1}$ die orthogonale Projektionsmatrix auf $\{\widehat{Z}_i\}_1^{j-1}$ ist. Mit $\widehat{Z}_{j\cdot 1,\dots,j-1}$ bezeichnen wir also die Residuen nach Anpassung von $\widehat{Z}_j$ durch $\widehat{Z}_1,\dots,\widehat{Z}_{j-1}$. Man beachte, dass (4.6) von der Reihenfolge der $\widehat{Z}_i$ abhängt. Aufgrund der natürlichen Ordnung bei der Hauptkomponentenanalyse stellt dies aber kein Problem dar. Somit ergibt sich die Gesamtvarianz der ersten k Hauptkomponenten durch

$$\sum_{j=1}^{k}\left\|\widehat{Z}_{j\cdot 1,\dots,j-1}\right\|^2. \tag{4.7}$$

Mithilfe einer QR-Zerlegung von $\widehat{\mathbf{Z}} = \mathbf{QR}$, wobei $\mathbf{Q}$ orthonormal und $\mathbf{R}$ eine recht obere Dreiecksmatrix ist, können wir (4.7) schnell berechnen, denn $\left\|\widehat{Z}_{j\cdot 1,\dots,j-1}\right\|^2 = R_{jj}^2$. Auch wenn dieser Ansatz zunächst sinnvoll scheinen mag, werden wir in Kapitel 6 anhand unseres Datensatzes zeigen, dass durch (4.7) keine korrekte Berechnung der erklärten Varianz erfolgt. Das Problem des Ansatzes liegt darin, dass der Bezug zum Rekonstruktionsfehler unklar ist. Anders als bei der klassischen Hauptkomponentenanalyse stimmen erklärte Varianz und Rekonstruktionsfehler bei der dünnbesetzten Variante nicht mehr überein.

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

$$\begin{aligned}
\|\mathbf{X}\|_F^2 &= \left\|\widehat{\mathbf{Z}}\widehat{\mathbf{B}}^\top + \widehat{\mathbf{E}}\right\|_F^2 \\
&= \operatorname{tr}\left(\widehat{\mathbf{B}}\widehat{\mathbf{Z}}^\top\widehat{\mathbf{Z}}\widehat{\mathbf{B}}^\top\right) + \operatorname{tr}\left(\widehat{\mathbf{B}}\widehat{\mathbf{Z}}^\top\widehat{\mathbf{E}}\right) + \operatorname{tr}\left(\widehat{\mathbf{E}}^\top\widehat{\mathbf{Z}}\widehat{\mathbf{B}}^\top\right) + \operatorname{tr}\left(\widehat{\mathbf{E}}^\top\widehat{\mathbf{E}}\right) \\
&= \operatorname{tr}\left(\widehat{\mathbf{B}}\widehat{\mathbf{Z}}^\top\widehat{\mathbf{Z}}\widehat{\mathbf{B}}^\top\right) + \operatorname{tr}\left(\widehat{\mathbf{E}}^\top\widehat{\mathbf{E}}\right) \\
&= \left\|\widehat{\mathbf{Z}}\widehat{\mathbf{B}}^\top\right\|_F^2 + \left\|\widehat{\mathbf{E}}\right\|_F^2
\end{aligned} \tag{4.8}$$

Damit kann die Varianz eines Datensatzes in Rekonstruktion und Residuum aufgeteilt werden. Ersterer Teil entspricht der erklärten Varianz unseres Modells und wird daher mit $\left\|\widehat{\mathbf{Z}}\widehat{\mathbf{B}}^\top\right\|_F^2$ 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 λ_2. Im Folgenden möchten wir verschiedene Vorgehensweisen 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 vorgehen. Aufgrund der Recheneffizienz bestimmen wir k aber meist durch das Ergebnis der klassischen Variante.

Empirische Studien zeigen, dass sich die Ergebnisse bei Veränderung von λ_2 kaum ändern. Ist $n > p$ für unseren Datensatz, können wir den Hyperparameter auf Null setzen, da das Lasso-Defizit in diesem Fall nicht auftritt. In der Praxis wird λ_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 spezielle Wahl von λ_2 in Kapitel 5 treffen.

Komplizierter gestaltet sich eine Wahl von $\lambda_{1,j}$, welche die Modellkomplexität wesentlich beeinflusst. 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. Folgende Variante beruht auf [Hub+16; AM11]

$$\mathrm{BIC}(\lambda_{1,j}) = \log\left(\frac{\left\|X - Z_j\beta_j^\top\right\|_F^2}{np}\right) + \mathrm{df}(\lambda_{1,j})\frac{\log(np)}{np}, \tag{4.9}$$

wobei $\mathrm{df}(\lambda_{1,j}) = \|\beta_j\|_0$ die Anzahl von null verschiedener Einträge sind. Dabei steht df für *degrees of freedom*, welches die Anzahl freier Parameter darstellt [HTF09]. Klar erkennbar in (4.9) ist der Kompromiss zwischen Rekonstruktionsfehler der j-ten Hauptkomponente und der Dünnbesetzung durch $\mathrm{df}(\lambda_{1,j})$. Mit steigendem $\lambda_{1,j}$ wird der Rekonstruktionsfehler größer und die Anzahl freier Parameter geringer. Wir sind angehalten $\lambda_{1,j}$ zu finden, welche eine Balance zwischen den beiden Termen ermöglichen.

Um nicht über k Hyperparameter optimieren zu müssen, kann man $\lambda_{1,j} = \lambda_1$ für alle $1 \leq j \leq k$ setzen. Eine weitere Möglichkeit wird in [CFF13] beschrieben. Für $j > 1$ sei $\widehat{\mathbf{B}}_{j-1}^\perp$ die Matrix, deren Spalten eine orthonormale Basis für das orthogonale Komplement für den durch $\widehat{\beta}_1, \ldots, \widehat{\beta}_{j-1}$ aufgespannten Raum sind. Wir berechnen $x_i^{(j-1)} = (\widehat{\mathbf{B}}_{j-1}^\perp)^\top x_i$ für $i = 1, \ldots, n$ und setzen $\lambda_{1,j} = \lambda_1 \operatorname{Var}[\mathbf{X}^{(j)}]$, 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 erreichen. Nun

muss lediglich λ_1 gewählt werden. Hierfür wird von [CFF13; Guo+10] ein ähnliches BIC-Kriterium vorgeschlagen

$$\mathrm{BIC}(\lambda_1) = \frac{\left\|\mathbf{X} - \mathbf{X}\widehat{\mathbf{B}}\widehat{\mathbf{A}}^\top\right\|_F^2}{\left\|\mathbf{X} - \mathbf{X}\widehat{\mathbf{V}}\widehat{\mathbf{V}}^\top\right\|_F^2} + \mathrm{df}(\lambda_1)\frac{\log(n)}{n}. \tag{4.10}$$

Für die log-likelihood-Funktion wird in (4.10) das Verhältnis des Rekonstruktionsfehlers zwischen dünnbesetzter und der klassischer Hauptkomponentenanalyse gewählt. Welche der beiden BIC-Kriterien genutzt werden sollte, kommt auf den Anwendungsfall an.

Typischerweise wird für die Minimierung der BIC-Kriterien eine Rastersuche für λ_1 im Wertebereich $[0, \lambda_1^{max}]$ durchgeführt, wobei eine Wahl von λ_1^{max} in Hauptachsen mit nur einem von Null verschiedenem Eintrag resultieren. Andere Suchverfahren wie die Zufallssuche, Bayessche oder gradientenbasierte Optimierung sind an dieser Stelle denkbar.

4.6 Theoretische Aussagen

Zum 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 für hochdimensionale Datensätze.

In Kapitel 5 werden wir sehen, dass das Sparse PCA Kriterium nur von der Kovarianzmatrix abhängt. Um eine Populationsversion der dünnbesetzen Hauptkomponete zu erhalten, ersetzen wir wie zuvor $\mathbf{X}^\top\mathbf{X}$ durch die Kovarianzmatrix $\mathbf{\Sigma}$. Weil eine Betrachtung einer *high dimension low sample size (HDLSS)* Situation die Einführung zu vieler neuer Prinzipien erfordern würde, präsentieren wir ein Konsistenz-Resultat von Johnstone und Lu [JY09], falls die Hauptkomponentenanalyse auf eine Teilmenge der Variablen angewendet wird. Dies liefert eine theoretische Rechtfertigung der dünnbesetzten Variante. Dazu wenden wir uns wieder einem *single component model* zu

$$x_i = l_j v_1 + \sigma r_i \quad i = 1, \dots n, \tag{4.11}$$

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 $r_i \sim N_p(0, \mathbb{1})$ unabhängige Rauschvektoren. Wir bezeichnen mit $\widehat{I} = \{j : \widehat{\sigma}_j^2 \geq \sigma^2(1+\alpha_n)\}$ die Indexmenge der zu selektierenden Variablen, wobei $\widehat{\sigma}_j^2$ die Stichprobenvarianz der j-ten Variable und $\alpha_n = \alpha(n^{-1}\log(\max\{n,p\}))^{\frac{1}{2}}$ für ein hinreichend großes α ist. Da die großen Einträge von $\widehat{v}_1$ mit Variablen großer Varianz korrespondieren, wählen wir

$$\widehat{v}_{1j}^I = \begin{cases} \widehat{v}_{1j} & j \in \widehat{I} \\ 0 & j \notin \widehat{I} \end{cases}$$

. Mit dieser Selektion der Variablen erhalten wir die Konsistenz für hochdimensionale Fälle unter geeigneten Modellannahmen.

Theorem 4.2 (Konsistenz der dünnbesetzten Hauptkomponentenanalyse [JY09])**.** *Für $n \to \infty$ sei die Signalstärke $\|v_1\|^2 \to \rho$ asymptotisch stabil und $n^{-1}\log(\max\{n,p\}) \to 0$. Bezeichne mit $v_i^{(r)}$ den r-größten Eintrag von v_i. Unter der Annahme, dass die Einträge der Hauptachsen für alle n schnell abfallen, d.h.*

$$|v_1^{(r)}| \leq C r^{\frac{-1}{q}} \quad \textit{für } q \in (0,2) \textit{ und } c < \infty$$

ist die dünnbesetzte Hauptkomponentenanalyse fast sicher konsistent

$$\mathbb{P}\left[\lim_{n\to\infty} \text{angle}(\widehat{v}_1^I, v_1) = 0\right] = 1.$$

In einer HDLSS-Situation zeigen Shen et al. die Konsistenz einer ähnlichen Variante von Sparse PCA mithilfe eines *spiked covariance models* [SH08]. Der einzige Unterschied zu dem Ansatz von Zou et al. ist, dass die Hauptachsen dort sequentiell statt simultan berechnet werden. Für Details bezüglich der speziellen Bedingungen für eine Konsistenz verweisen wir auf [SSM13].

Kapitel 5

Implementierung

In diesem Kapitel werden wir einen Algorithmus beschreiben, der das Sparse PCA Kriterium (4.3) minimiert. Dabei sehen wir uns mit einem nicht-konvexem Optimierungsproblem konfrontiert. Allerdings können wir ausnutzen, dass (4.3) konvex bezüglich $\widehat{\mathbf{A}}$ bzw. $\widehat{\mathbf{B}}$ ist, falls wir eine der beiden Matrizen fixieren. Im Folgenden werden wir zunächst eine allgemeine numerische Lösung präsentieren und die Komplexität des assoziierten Algorithmus bestimmen. In einer HDLSS-Situation werden wir diesen in 5.4 leicht abändern, um eine effiziente Berechnung zu garantieren. Zum Schluss dieses Kapitels diskutieren wir Details einer eigenen Implementierung in Python.

5.1 Numerische Lösung

Für eine einfache Übersicht werden wir das Sparse PCA Kriterium hier wiederholen.

$$(\widehat{\mathbf{A}}, \widehat{\mathbf{B}}) = \arg\min_{\mathbf{A},\mathbf{B}} \sum_{i=1}^{n} \left\| x_i - \mathbf{A}\mathbf{B}^\top x_i \right\|_2^2 + \lambda_2 \sum_{j=1}^{k} \|\beta_j\|_2^2 + \sum_{j=1}^{k} \lambda_{1,j} \|\beta_j\|_1 \tag{5.1}$$
$$\text{unter der Nebenbedingung, dass } \mathbf{A}^\top \mathbf{A} = \mathbb{1}_{k\times k}.$$

Wie im Einstieg erwähnt handelt es sich bei (5.1) um ein bikonvexes Optimierungsproblem. Daher liegt es nahe einen alternierenden Ansatz zu wählen, um das Problem numerisch zu lösen. Somit fixieren wir im Folgenden eine der beiden Matrizen.

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

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

Somit reduziert sich (5.1) für fixes $\mathbf{A}$ auf das Lösen von k Elastic Net Problemen

$$\widehat{\beta}_j = \arg\min_{\beta_j} \left\| Y_j^* - \mathbf{X}\beta_j \right\|_2^2 + \lambda_2 \left\| \beta_j \right\|_2^2 + \lambda_{1,j} \left\| \beta_j \right\|_1, \tag{5.2}$$

wobei $Y_j^* = \mathbf{X}\alpha_j$ für alle $1 \leq j \leq k$. In Abschnitt 2.2.6 haben wir uns ausführlich mit Elastic Nets beschäftigt und ein effizientes Koordinaten-Abstiegsverfahren zur Lösung dieser präsentiert.

A gegeben B:

Fixieren wir die Matrix $\mathbf{B}$, können wir uns auf das Minimieren der Verlustfunktion $\left\| \mathbf{X} - \mathbf{X}\mathbf{B}\mathbf{A}^\top \right\|_F^2$ beschränken, da die Bedingungen an β_j nicht von Relevanz beim Optimieren über $\mathbf{A}$ sind. Somit reduziert sich (5.1) für fixes $\mathbf{B}$ auf das Lösen von

$$\begin{gathered} \widehat{\mathbf{A}} = \arg\min_{\mathbf{A}} \left\| \mathbf{X} - \mathbf{X}\mathbf{B}\mathbf{A}^\top \right\|_F^2 \\ \text{unter der Nebenbedingung, dass } \mathbf{A}^\top\mathbf{A} = \mathbb{1}_{k\times k}. \end{gathered} \tag{5.3}$$

Für dieses Optimierungsproblem lässt sich eine explizite Lösung angeben. Es ist eine Form von Procrustes Rotationsproblem, welches wir ebenfalls in Kapitel 2 beschrieben haben. Die Matrix $\mathbf{X}\mathbf{B}$ soll durch Multiplikation mit einer orthogonalen Matrix $\mathbf{A}$ in $\mathbf{X}$ überführt werden. Berechnen wir eine Singulärwertzerlegung von $(\mathbf{X}^\top\mathbf{X})\mathbf{B} = \mathbf{U}\mathbf{D}\mathbf{V}^\top$, ist die Lösung für (5.3) gegeben durch

$$\widehat{\mathbf{A}} = \mathbf{U}\mathbf{V}^\top. \tag{5.4}$$

5.2 Algorithmus

Durch die Vorarbeit im vorangegangenem Abschnitt können wir einen effizienten Algorithmus zur Lösung des Sparse PCA Kriteriums angeben. Zuerst initialisieren wir $\mathbf{A}$ mit den ersten k gewöhnlichen Hauptachsen. Anschließend minimieren wir abwechselnd über die Matrizen $\mathbf{A}$ und $\mathbf{B}$ 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 2 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 die Iteration in ihrer Implementierung 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 β_i in der l-ten Iteration ist. Sobald also die Änderung in $\mathbf{B}$ klein genug ist, kann die while-Schleife beendet 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 2 Sparse Principal Component Analysis

1: **procedure** SPCA($\mathbf{A}, \mathbf{B}, k, \lambda_2, \lambda_{1,j}$)
2: $\mathbf{A} \leftarrow \mathbf{V}[, 1:k]$, die ersten k Hauptachsen
3: **while** nicht konvergiert **do**
4: Gegeben festes $\mathbf{A} = [\alpha_1, \ldots, \alpha_k]$, löse das elastic net Problem

$$\widehat{\beta}_j = \arg\min_{\beta_j} \left\|\mathbf{X}\alpha_j - \mathbf{X}\beta_j\right\|^2 + \lambda_2 \left\|\beta_j\right\|^2 + \lambda_{1,j} \left\|\beta_j\right\|_1 \quad \text{für } j = 1, \ldots, k$$

5: Gegeben festes $\mathbf{B} = [\beta_1, \ldots, \beta_k]$, berechne die Singulärwerzerlegung von

$$\mathbf{X}^\top \mathbf{X}\mathbf{B} = \mathbf{U}\mathbf{D}\mathbf{V}^\top$$

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

6: $\widehat{V}_j \leftarrow \frac{\beta_j}{\|\beta_j\|}$ for $j = 1, \ldots, k$

5.3 Komplexität

Wir werden uns nun mit der Komplexität von Algorithmus 2 beschäftigen. Dabei werden wir wieder einmal zwischen den Fällen $n > p$ und $p \gg n$ unterscheiden.

Fall: n > p

In diesem Fall lässt sich ein Trick für die Berechnung von (5.2) anwenden. Indem wir

$$\begin{aligned}\widehat{\beta}_j &= \arg\min_{\beta_j} \left\|Y_j^* - \mathbf{X}\beta_j\right\|_2^2 + \lambda_2 \left\|\beta_j\right\|_2^2 + \lambda_{1,j} \left\|\beta_j\right\|_1 \\ &= \arg\min_{\beta_j} (\alpha_j - \beta_j)^\top \mathbf{X}^\top \mathbf{X}(\alpha_j - \beta_j) + \lambda_2 \left\|\beta_j\right\|_2^2 + \lambda_{1,j} \left\|\beta_j\right\|_1 \end{aligned} \tag{5.5}$$

umformen, hängen beide Subprobleme (5.2) und (5.3) nur von der Kovarianzmatrix $\mathbf{X}^\top\mathbf{X}$ ab. Daher ist es sinnvoll, diese vorab zu berechnen, um die Anzahl an Multiplikationen je Iteration zu verringern. Allerdings ist (5.5) mit $\mathbf{X}^\top\mathbf{X}$ nicht direkt ein Elastic Net Problem. Definieren wir $Y^{**} = (\mathbf{X}^\top\mathbf{X})^{\frac{1}{2}}\alpha_j$ und $\mathbf{X}^{**} = (\mathbf{X}^\top\mathbf{X})^{\frac{1}{2}}$ können wir (5.5) aber in ein Elastic Net Problem transformieren

$$\widehat{\beta}_j = \arg\min_{\beta} \|Y^{**} - \mathbf{X}^{**}\beta\|_2^2 + \lambda_2 \|\beta\|_2^2 + \lambda_{1,j} \|\beta\|_1 .$$

Um die Laufzeit des Algorithmus zu bestimmen, wenden wir uns Tabelle 5.1 zu, welche die verschiedenen Berechnungsschritte zeigt. Die Initialisierung von $\mathbf{A}$ durch die ersten k gewöhnlichen Hauptachsen wird durch eine abgeschnittene Singulärwertzerlegung von $\mathbf{X}$ berechnet. Zudem berechnen wir vorab die Kovarianzmatrix $\mathbf{X}^\top\mathbf{X}$. Pro Iteration lösen wir k Elastic Net Probleme und ein Procrustes Rotationsproblem. Insgesamt ergibt sich aufgrund $k < \min\{n, p\}$ und $j < p$ eine Laufzeit von $np^2 + mk\mathcal{O}(p^3)$, wobei m die Anzahl der Iterationen ist.

Berechnung	Komplexität
Singulärwertzerlegung von $\mathbf{X}$	$\mathcal{O}(npk)$
$\mathbf{X}^\top\mathbf{X}$	$\mathcal{O}(np^2)$
$(\mathbf{X}^\top\mathbf{X})\mathbf{B}$	$\mathcal{O}(p^2k)$
$\mathbf{X}^\top(\mathbf{XB})$	$\mathcal{O}(npk)$
Singulärwertzerlegung von $\mathbf{X}^\top\mathbf{XB}$	$\mathcal{O}(pk^2)$
Elastic Net Problem	$\mathcal{O}(pnj + j^3)$

TABELLE 5.1: Die Tabelle zeigt die Komplexität für die in Algorithmus 2 vorkommenden Berechnungen. Dabei ist j die Anzahl von Null verschiedener Koeffizienten.

Fall: $\mathbf{p \gg n}$

Für diesen Fall ist eine Berechnung der Kovarianzmatrix $\mathbf{X}^\top\mathbf{X}$ nicht mehr sinnvoll, da dies eine $p \times p$-Matrix ist. Daher werden wir $\mathbf{X}^\top(\mathbf{XB})$ in jeder Iteration naiv berechnen. Die Laufzeit des Algorithmus wird in diesem Fall von der Lösung der k Elastic Nets dominiert, besonders wenn wir viele von Null verschiedene Einträge zulassen. Insgesamt ergibt sich die Komplexität $mk\mathcal{O}(pnj + j^3)$. Für große j bzw. p können die Berechnungskosten somit sehr hoch sein, weshalb wir im folgenden Abschnitt einen speziellen Sparse PCA Algorithmus kennenlernen werden.

5.4 Numerische Lösung im Fall $\mathbf{p \gg n}$

Für viele Anwendungen kann die Anzahl an Variablen die Anzahl an Beobachtungen deutlich übersteigen. Um auch in diesem Fall eine effiziente Berechnung zu ermöglichen, formulieren wir einen Spezialfall des Algorithmus 2.

Dazu beobachten wir, dass Theorem 4.1 für alle $\lambda_2 > 0$ gilt. Für $\lambda_2 \to \infty$ erhalten wir eine interessante Charakterisierung.

Theorem 5.1. *Seien* $\frac{\widehat{\beta}_j(\lambda_2)}{\|\widehat{\beta}_j(\lambda_2)\|}$ *die dünnbesetzten Hauptachsen aus (4.3) in Abhängigkeit von* λ_2 *und* $(\tilde{\mathbf{A}}, \tilde{\mathbf{B}})$ *die Lösung des Optimierungsproblems*

$$(\tilde{\mathbf{A}}, \tilde{\mathbf{B}}) = \underset{\mathbf{A},\mathbf{B}}{\arg\min} -2tr\left(\mathbf{A}^\top\mathbf{X}^\top\mathbf{XB}\right) + \sum_{j=1}^{k} \|\beta_j\|_2^2 + \sum_{j=1}^{k} \lambda_{1,j} \|\beta_j\|_1 \tag{5.6}$$
$$\textit{unter der Nebenbedingung, dass } \mathbf{A}^\top\mathbf{A} = \mathbb{1}_{k\times k}.$$

Für $\lambda_2 \to \infty$ *konvergieren die dünnbesetzten Hauptachsen* $\frac{\widehat{\beta}_j(\lambda_2)}{\|\widehat{\beta}_j(\lambda_2)\|} \to \frac{\tilde{\beta}_j}{\|\tilde{\beta}_j\|}$ *gegen die Lösung von ().*

Daher können wir das vereinfachte Optimierungsproblem (5.1) benutzen, um das Sparse PCA Kriterium für den Spezialfall $\lambda_2 = \infty$ zu lösen. Fixieren wir wie in Algorithmus 2 die Matrix $\mathbf{A}$, verbleiben wir mit dem Problem

$$\widehat{\beta}_j = \underset{\beta_j}{\arg\min} -2\alpha_j^\top(\mathbf{X}^\top\mathbf{X})\beta_j + \|\beta_j\|_2^2 + \lambda_{1,j}\|\beta_j\|_1 . \tag{5.7}$$

Für (5.7) können wir aufgrund des Wegfalls von λ_2 eine explizite Lösung angeben

$$\widehat{\beta}_j = \text{soft}_{\frac{\lambda_{1,j}}{2}}(\alpha_j^\top \mathbf{X}^\top \mathbf{X}) = \left(|\alpha_j^\top \mathbf{X}^\top \mathbf{X}| - \frac{\lambda_{1,j}}{2} \right)_+ \text{Sign}(\alpha_j^\top \mathbf{X}^\top \mathbf{X}). \tag{5.8}$$

Daher ersetzen wir die Berechnung in Schritt 4 von Algorithmus 2 durch (5.8) falls $p \gg n$. Für den hochdimensionalen Fall kann durch das Wegfallen der Berechnung von k Elastic Net Problemen somit ein effizientes Verfahren gewährleistet werden.

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 (5.2) wird der LARS-EN Algorithmus gewählt, welche eine Erweiterung des LARS-Algorithmus für das Elastic Net ist [ZH05]. Dagegen bietet scikit-learn eine Sparse PCA-Variante in Python, welche auf [JOB10] zurückgeht und einen 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 korrekte Ergebnisse erzielt. 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 (5.2) mit LARS-EN zu lösen, wählen wir ein randomisiertes Koordinaten-Abstiegsverfahren, welches wir in Abschnitt 2.2.6 beschrieben haben. Des Weiteren wird in der Implementierung von Zou et al. die Gram-Matrix $\mathbf{X}^\top \mathbf{X}$ immer vorab berechnet, um für das Subproblem (5.3) nur eine Multiplikation pro Iteration $(\mathbf{X}^\top \mathbf{X})\mathbf{B}$ durchführen zu müssen. Da die Gram-Matrix aber in $\mathbb{R}^{p \times p}$ liegt, ist es für den Fall $p \gg n$ sinnvoller, $\mathbf{X}^\top(\mathbf{X}\mathbf{B})$ 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 2.2.7 definieren wir

$$\lambda = \frac{2\lambda_2 + \lambda_1}{2n} \quad \text{und} \quad \alpha = \frac{\lambda_1}{2\lambda_2 + \lambda_1} \tag{5.9}$$

wobei λ die Stärke der Bestrafung und α das Verhältnis des ℓ_1 zum ℓ_2-Strafterm beschreibt.

Kapitel 6

Anwendung

In diesem Kapitel beschäftigen wir uns mit der Anwendung der dünnbesetzten Hauptkomponentenanalyse auf Frequenzdaten einer Mühle. Dafür stehen uns Zeitreihen von Beschleunigungssensoren und Mikrofonen zur Verfügung, welche an der Maschine angebracht sind, um die Vibration bzw. die Akustik zu messen. Wir sind interessiert daran herauszufinden, ob sich mithilfe der Zeitreihen Aussagen über die Partikelgröße des Materials treffen lassen. Aufgrund der geringen Beobachtungszahl des Datensatzes sind viele überwachte Lernverfahren in diesem Zusammenhang unbrauchbar. Im Zuge einer explorativen Analyse kann daher eine Dimensionsreduktion sinnvoll sein. Dabei sind wir nicht unbedingt an der niedrigdimensionalen Repräsentation der Daten interessiert, sondern viel mehr an der Herausfilterung wichtiger Frequenzen. Idealerweise können wir die Frequenzen dem Material oder der Maschine zuordnen, so dass wir zwischen Material- und Maschineneigenschaften unterscheiden können. Die Möglichkeit der Interpretation und die Konsistenz in hochdimensionalen Fällen waren Anlass für die Verwendung der dünnbesetzten Hauptkomponentenanalyse in diesem Zusammenhang.

Zunächst werden wir den Datensatz in Abschnitt 6.1 näher beschreiben und Vorverarbeitungsschritte in 6.2 erläutern. Nachdem wir in 6.3 erklären, welche Experimente durchgeführt worden sind, werden wir uns den Ergebnissen in 6.4 zuwenden. Zu einer detaillierten Auswertung gehört sowohl ein Vergleich mit der klassischen Hauptkomponentenanalyse, als auch eine Analyse des Verhalten des Algorithmus. Dazu werden wir die Wahl der Hyperparameter, die Laufzeit und die Konvergenz thematisieren. Zum Abschluss dieses Kapitels werden wir die Korrektheit der von Camacho et al. [Cam+20] berechneten Varianzen empirisch validieren. Ob eine derartige Methode für diesen Datensatz ein sinnvoller Ansatz war, diskutieren wir in Kapitel 7.

6.1 Beschreibung des Datensatzes

Wir verfügen über Zeitreihen verschiedener Sensoren, die an unterschiedlichen Positionen einer Mühle angebracht sind. Insgesamt wurden $n \approx 30$ Messungen bei laufendem Mahlprozess gestartet. Bei jeder dieser wurde dasselbe Material verwendet und ein festgelegter Zeitraum betrachtet. Um eine Trennung von Maschine und Material in den Daten zu ermöglichen, wurden Messungen sowohl mit als auch ohne Material durchgeführt. Des Weiteren wurden verschiedene Betriebszustände der Maschine variiert, damit ein möglicher Zusammenhang mit dem Mahlergebnis hergestellt werden kann. Durch eine hohe Abtastrate haben wir es mit einem hochdimensionalen Datensatz zu tun. Für jeden angebrachten Sensor erhalten wir Auslenkungswerte für $p \approx 5,000,000$ Zeitpunkte. Mit einer geringen Beobachtungszahl, wobei Messungen mit Material mehrmals aufgezeichnet worden sind, sehen wir uns mit einer *high dimension low sample size* Situation konfrontiert.

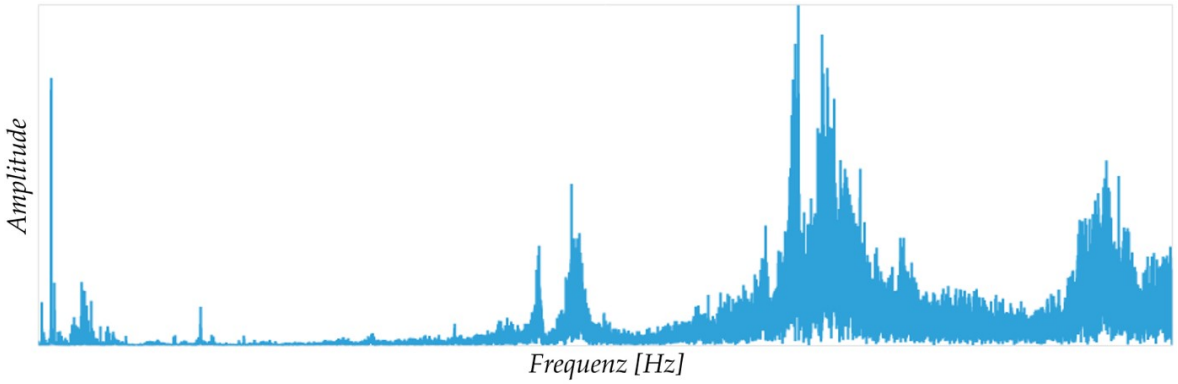


ABBILDUNG 6.1: Ergebnis einer Fouriertransformation für einen Beschleunigungssensor.

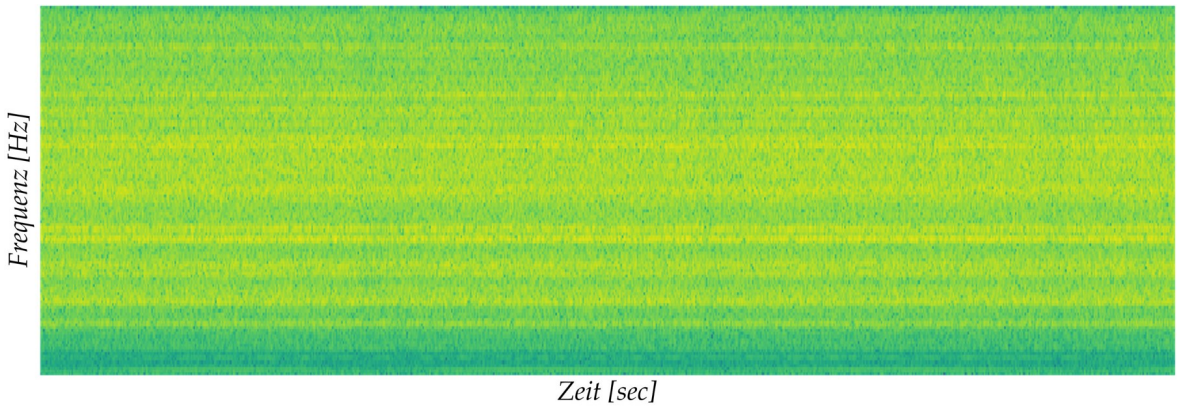


ABBILDUNG 6.2: Zeitlicher Verlauf der beobachteten Frequenzen.

6.2 Vorverarbeitung der Daten

Vor der Anwendung der dünnbesetzten Hauptkomponentenanalyse haben wir einige Vorverarbeitungsschritte vorgenommen. Da die Messungen zu zufälligen Zeitpunkten bei laufendem Mahlprozess gestartet worden sind, können einzelne Zeitpunkte nicht direkt miteinander verglichen werden. Mit einer Fouriertransformation der Daten können wir anstatt auf der Zeitachse auf den Frequenzen arbeiten, welche uns tiefere Einblicke ermöglichen. Auf diesem Wege sind sowohl Unterschiede zwischen Messungen, als auch mögliche Rauscheffekte leichter zu erkennen. In Abbildung 6.1 zeigen wir das Ergebnis einer Fouriertransformation beispielhaft für eine Messung eines Beschleunigungssensors.

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. Mithilfe eines Blicks auf das Spektrogramm in Abbildung 6.2 erkennen wir anhand der horizontalen Linien, dass sich die Frequenzen über den Messzeitraum kaum verändern. Dies ist darauf zurückzuführen, dass der Maschine konstant Material zugeführt wird. Daher 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. Somit verbleiben wir mit $p \approx 15,000$ Variablen. Um die Varianzen der Frequenzen vergleichbarer zu machen, haben wir die Daten ähnlich wie bei der klassischen Hauptkomponentenanalyse zentriert.

6.3 Anwendung auf Frequenzdaten

Unsere Implementierung ermöglicht eine Wahl verschiedener Parameter bezüglich des Modells bzw. des Algorithmus. 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 Modellparameter, den es zu wählen gilt, ist die Anzahl zu berechnender Hauptkomponenten. Wie bereits in 4.5 beschrieben, wird dazu meist das Ergebnis einer klassischen Hauptkomponentenanalyse verwendet. Hierbei hat sich gezeigt, dass 2 Hauptkomponenten für die meisten Sensoren ausreichen, um einen Großteil des Datensatzes zu erklären. Zu Analysezwecken haben wir uns entschieden einen Durchlauf mit 2 und einen mit 10 Hauptkomponenten zu starten.

Interessanter ist die Wahl der Hyperparameter λ und α, welche wesentlichen Einfluss auf die Ergebnisse besitzen. Auch wenn prinzipiell die Möglichkeit besteht, differenzierte Bestrafungen je Hauptkomponente $\lambda_{1,j}$ auszuwählen, beschränken wir uns der Einfachheit halber auf eine einheitliche Bestrafung. Für die Wahl von λ haben wir sowohl mehrere Werte ausprobiert, als auch eine Rastersuche bezüglich der in 4.5 beschriebenen BIC-Kriterien durchgeführt. Dafür verwenden wir auf einer log-Skala gleichverteilte Werte im Bereich zwischen 10^{-7} und 10^{0}. Dagegen wählen wir für das Verhältnis zwischen Lasso und Ridge-Bestrafung die Werte $[0.1, 0.5, 0.7, 0.9, 0.95, 0.99, 1]$. Mithilfe des BIC-Kriteriums können wir zeitgleich über λ und α optimieren. Es hat sich in den Anwendungen gezeigt, dass eine geeignete Liste für α mehr Werte nahe 1 hat, da sich dort die größten Änderungen ergeben. Damit stärken wir den Lasso-Strafterm im Vergleich zum Ridge-Strafterm.

Exemplarisch werden wir uns nun mit einem der Beschleunigungssensoren weiter beschäftigen. An dieser Stelle möchten wir erwähnen, dass wir die Sensoren getrennt betrachten, d.h. die dünnbesetzte Hauptkomponentenanalyse auf die Sensoren einzeln anwenden. Dies ist aufgrund der Unterschiedlichkeit der Sensoren sinnvoll. Wir werden nun ausgewählte Ergebnisse vorstellen.

6.4 Auswertung der Ergebnisse

Im Rahmen dieser Arbeit können wir nun einen begrenzten Teil der Ergebnisse präsentieren. Ziel wird es sein, wesentliche Ergebnisse und Effekte des Algorithmus zu erläutern, um ein transparentes Modell zu entwickeln.

6.4.1 Klassische Analyse der Hauptachsen

Zunächst wollen wir eine klassische Analyse durchführen, wie sie oft für ein derartiges Verfahren gemacht wird. Um einen Vergleich zu ermöglichen, haben wir sowohl die klassische als auch die dünnbesetzte Hauptkomponentenanalyse durchgeführt.

Für einen ersten Einblick in die Ergebnisse betrachten wir Abbildung 6.3a bzw. 6.3b, in welcher die ersten beiden Hauptkomponenten gegeneinander aufgezeichnet sind. Dies ist die Darstellung der Daten bezüglich der neu gefundenen Variablen. Zuerst wenden wir uns den Ergebnissen der klassischen Hauptkomponentenanalyse zu. Man sieht schnell, dass sich vier Gruppierungen ergeben. Durch die Transformation sehen wir eine deutliche Trennung zwischen Messungen mit und ohne Material, welche im Bild durch verschiedene Farben

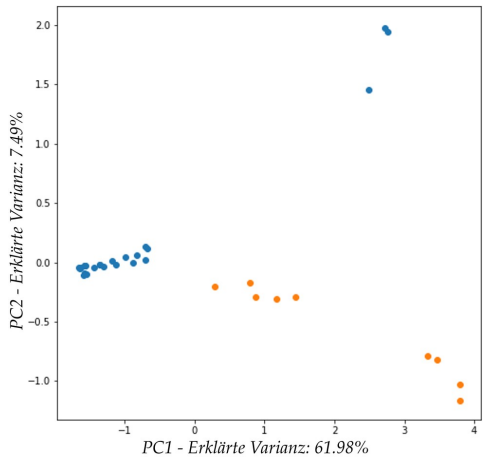
markiert sind. Des Weiteren sind die Messungen mit bzw. ohne Material in zwei Untergruppen geteilt. Die Unterschiede sind auf verschiedene Betriebszustände der Maschine zurückzuführen, auf welche wir hier nicht genauer eingehen können. Vergleichen wir dieses Ergebnis mit der dünnbesetzten Variante in Abbildung 6.3b, erkennen wir viele Gemeinsamkeiten. Es treten dieselben Gruppierungen wie zuvor auf, so dass noch immer eine Trennung der Messungen bezüglich der Befüllung möglich ist. Ein kleinerer Unterschied ist in der zweiten Hauptkomponenten zu sehen, da bei der dünnbesetzten Variante drei Messungen noch stärker separiert werden.

Nun gilt es, das Zustandekommen dieses Bildes zu erklären. Wir wollen verstehen, warum wir eine Trennung zwischen Messungen bezüglich der Befüllung sehen. Vor allem interessieren wir uns dafür, welche Frequenzen für diesen Unterschied verantwortlich sind. Zu diesem Zweck betrachten wir Abbildung 6.3c bzw. 6.3d, in welcher die Hauptachsen der beiden Varianten zu sehen sind. Anhand der Hauptachsen können wir erkennen, welche Frequenzen eine entscheidende Rolle bei der Erhaltung der maximalen Varianz spielen. Demnach sind die größten Unterschiede im Datensatz auf die Frequenzen mit der größten Amplitude zurückzuführen. Auch wenn es in den niederen Frequenzen nicht leicht zu erkennen ist, sind bei der klassischen Hauptachse alle $\approx 15,000$ Einträge von Null verschieden. Eine Erklärung der beschriebenen Effekte ist hier quasi unmöglich. Wir können lediglich aussagen, dass die höheren Frequenzen in einer gewissen Weise wichtig sind. Genau hier liegt das Problem der klassischen Variante, denn eine Interpretation der Hauptachsen ist selten aufschlussreich. Es fließen einfach zu viele Koeffizienten in das Modell ein. Dagegen reduziert sich bei der dünnbesetzten Variante die Anzahl von Null verschiedener Einträge auf ≈ 40 für $\lambda = 0.1$ und $\alpha = 0.1$. Durch die eingeschränkte Modellkomplexität erkennen wir drei Peaks im Frequenzspektrum. Im Wesentlichen sind also nur viel weniger Frequenzen für die beschriebene Trennung verantwortlich. Daher können wir folgern, dass genau diese Frequenzen durch die Maschine verursacht werden. Mit diesem Verfahren ist es also möglich Frequenzen zu identifizieren, welche eine klare Bedeutung im Kontext besitzen.

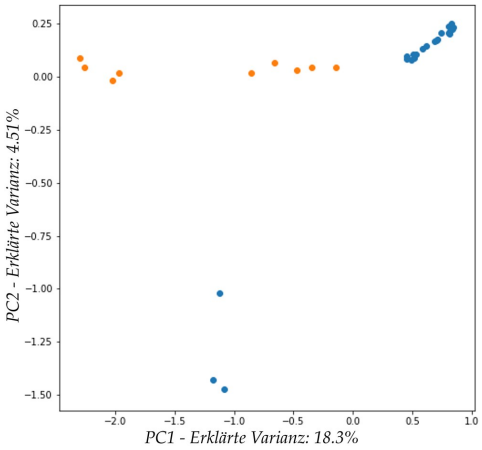
Zuletzt betrachtet man Abbildung 6.3e, welche einen sog. *scree plot* zeigt und oft als Bewertungsmittel von Dimensionsreduktionsverfahren dient. Mithilfe dieses Bildes kann man einsehen, welchen Anteil der Varianz des Datensatzes mit der niedrigdimensionalen Repräsentation erklärt wird. Zur Verdeutlichung der Effekte haben wir 10 Hauptkomponenten berechnet. Es zeichnet sich ein klarer Varianzverlust je Hauptkomponente ab, wenn wir die dünnbesetzte Hauptkomponentenanalyse nutzen. Hier erhalten wir nur 25% der Information des Datensatz, während es im Gegensatz dazu 95% bei der klassischen Variante sind. Besonders deutlich wird der Kompromiss, den wir eingehen müssen. Dadurch, dass unsere Hauptachsen einfacher zu interpretieren sind, verlieren wir an erklärter Varianz. Ein geeignetes Maß zwischen Modellkomplexität und Rekonstruktionsfehler zu finden, kann von der Anwendung und der Intention abhängen. In unserem Fall ist die Möglichkeit einer Interpretation der Hauptachsen deutlich wichtiger als die Varianzerhaltung. Wir haben durch Abbildung 6.3a bzw. 6.3b gesehen, dass uns durch einen Rückgang der Varianz nicht zwangsläufig Informationen verloren gehen müssen. Wir konnten trotz geringer Varianz ein ähnliches Bild mit denselben Gruppierungen erzielen.

6.4.2 Wahl der Hyperparameter

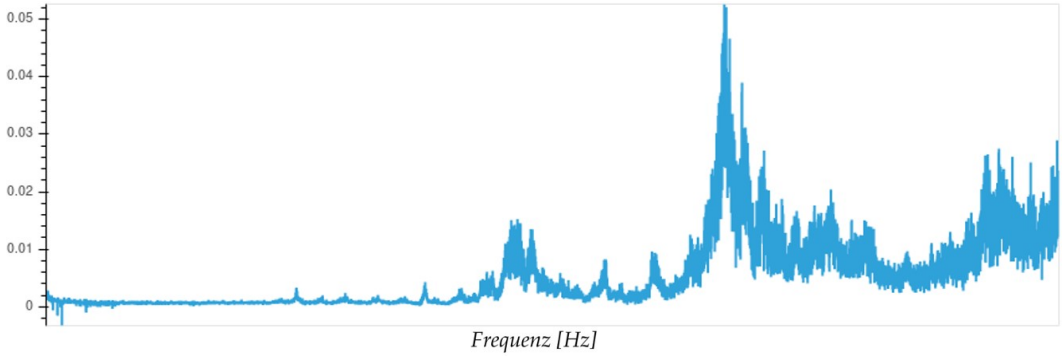
Im vorangegangenem Abschnitt haben wir beispielhaft gezeigt, wie die Ergebnisse einer dünnbesetzten Hauptkomponentenanalyse zu interpretieren sind. Für die Analyse haben wir bestimmte Werte von α und λ vorgegeben, die möglichst gute Ergebnisse erzielen. Es stellt sich jedoch die Frage, wie sich die einzelnen Hauptachsen und Hauptkomponenten



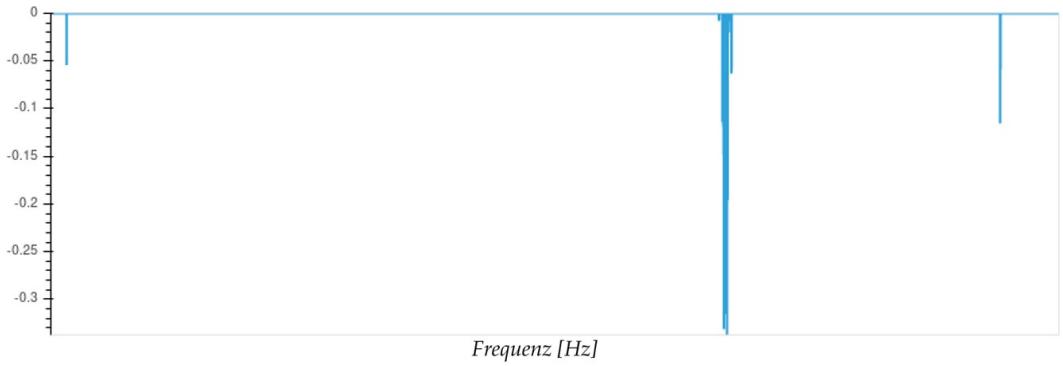
(A) Ergebnis einer klassische Hauptkomponentenanalyse.



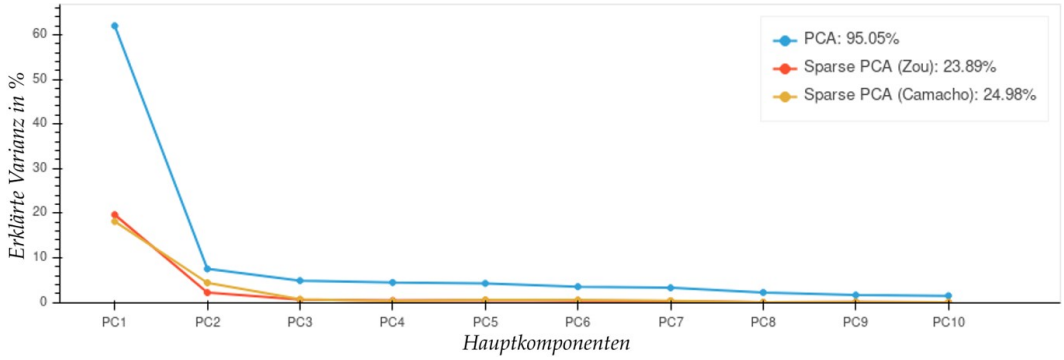
(B) Ergebnis einer dünnbesetzten Hauptkomponentenanalyse.



(C) Die Abbildung zeigt die Koeffizienten der ersten Hauptachse.



(D) Die Abbildung zeigt die Koeffizienten der ersten dünnbesetzten Hauptachse.



(E) Scree Plot für $k = 10$ Hauptkomponenten.

ABBILDUNG 6.3: Vergleich der klassischen Hauptkomponentenanalyse mit der dünnbesetzten Variante für $\lambda = 10^{-4}$ und $\alpha = 0.95$.

verändern, falls wir andere Werte für die Hyperparameter wählen. Zu diesem Zweck wenden wir uns Abbildung 6.4 zu. Hier haben wir die Anzahl an Hauptkomponenten fixiert und versucht, die Effekte in Abhängigkeit von λ für einen akustischen Sensor zu beschreiben.

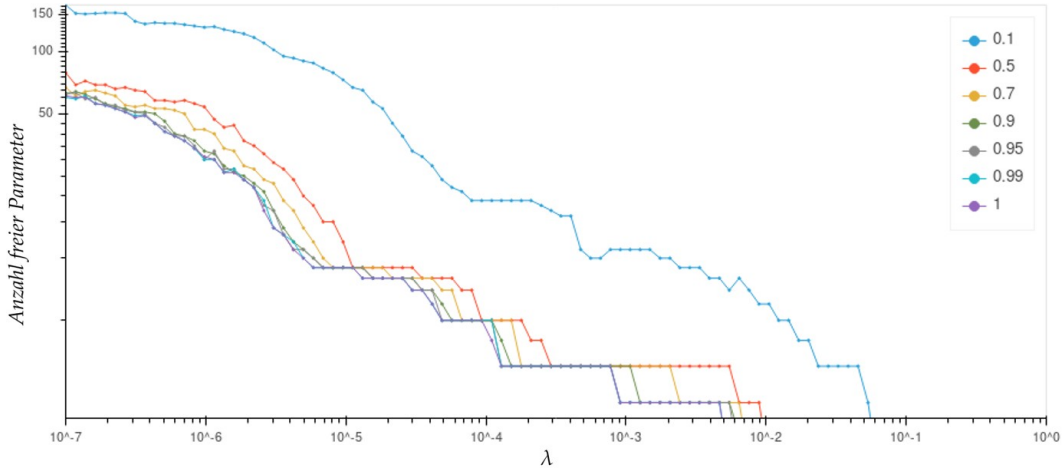
Abbildung 6.4a zeigt wie sich $\mathrm{df}(\lambda)$, also die Anzahl von Null verschiedener Einträge in den Hauptachsen, bei Veränderung von λ bzw. α verhält. Klar erkennbar ist der relativ gleichmäßige Abfall der Freiheitsgrade, d.h. für wachsendes λ werden unsere Hauptachsen zunehmend dünnbesetzt. Dies entspricht unseren Erwartungen aus Kapitel 4. Verschieben wir das Verhältnis der Bestrafung von der ℓ_1 zur ℓ_2-Norm, sprich kleineres α, steigt die Anzahl der Freiheitsgrade. Dies bestätigt, dass die ℓ_1-Norm im Gegensatz zur ℓ_2-Norm eine Dünnbesetzung hervorruft. Ab einem gewissem Punkt $\lambda \approx 10^{-2}$ für $\alpha > 0.5$ bzw. $\lambda \approx 10^{-1}$ für $\alpha = 0.1$ ist die Bestrafung zu stark, so dass die Hauptachsen dem Nullvektor entsprechen und keine Anpassung an den Datensatz mehr stattfindet.

Interessant ist nun, wie sich die erklärte Varianz des Datensatzes im Vergleich verhält, welche in Abbildung 6.4b zu sehen ist. Auffällig ist, dass sich für λ im Bereich von $[10^{-7}, 10^{-3}]$ kaum Änderungen in der Varianz ergeben. In diesem Bereich sind wir nur leicht unter dem Niveau der klassischen Variante. Im Umkehrschluss können wir aufgrund der kontinuierlichen Stagnation der Freiheitsgrade die Modellkomplexität verringern, aber zeitgleich den Rekonstruktionsfehler auf konstantem Niveau halten. Erst nahe $\lambda \approx 10^{-2}$ für $\alpha > 0.5$ bzw. bei $\lambda \approx 10^{-1}$ für $\alpha = 0.1$ zeichnet sich ein deutlicher Einbruch ab. Dieser ist dadurch zu erklären, dass die Hauptachsen dann dem Nullvektor entsprechen. Aus der Kombination der beiden Abbildungen können wir schließen, dass nur wenige Frequenzen wirklich wichtig zur Erklärung der Varianz des Datensatzes notwendig sind.

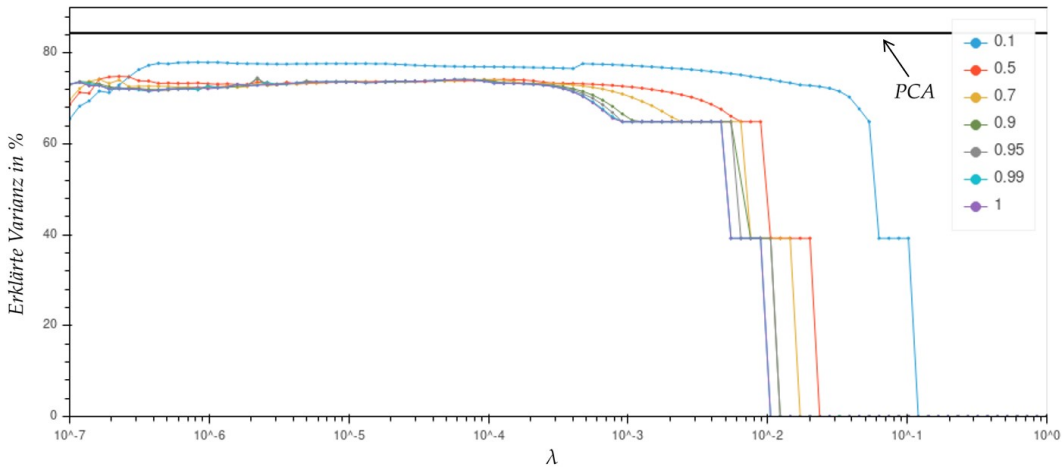
Um eine automatisierte Wahl von λ und α zu ermöglichen, haben wir in Abschnitt 4.5 Vorgehensweisen mithilfe eines BIC-Kriteriums beschrieben. Eine Anwendung des Kriteriums nach [CFF13; Guo+10] befindet sich in Abbildung 6.4c. Hier zeichnet sich ein Minimum im Bereich von $\lambda \in [10^{-4}, 10^{-3}]$ für $\alpha > 0.5$ bzw. nahe $\lambda \approx 10^{-2}$ für $\alpha = 0.1$ ab, welches wir nach obiger Analyse erwarten konnten. Es wird ein Punkt gewählt, an welchem die erklärte Varianz gerade noch auf sehr hohem Niveau, aber die Modellkomplexität gering ist. Letztere Abbildung ist also eine Kombination der Erkenntnisse und kann genutzt werden, um eine Balance zwischen Dünnbesetzung und erklärter Varianz zu finden. An dieser Stelle möchten wir erwähnen, dass die Resultate des BIC-Kriteriums nicht für alle Sensoren zufriedenstellend waren. Genauer gesagt sehen wir in manchen Fällen, dass die Gewichtung zwischen Modellkomplexität und Rekonstruktionsfehler nicht passend gewählt ist. Dies kann daran liegen, dass BIC-Kriterien in hochdimensionalen Fällen versagen können [Gir15]. Für unsere Zwecke haben wir daher die betroffenen Sensoren mit manuellen Gewichten korrigiert.

6.4.3 Verhalten des Algorithmus

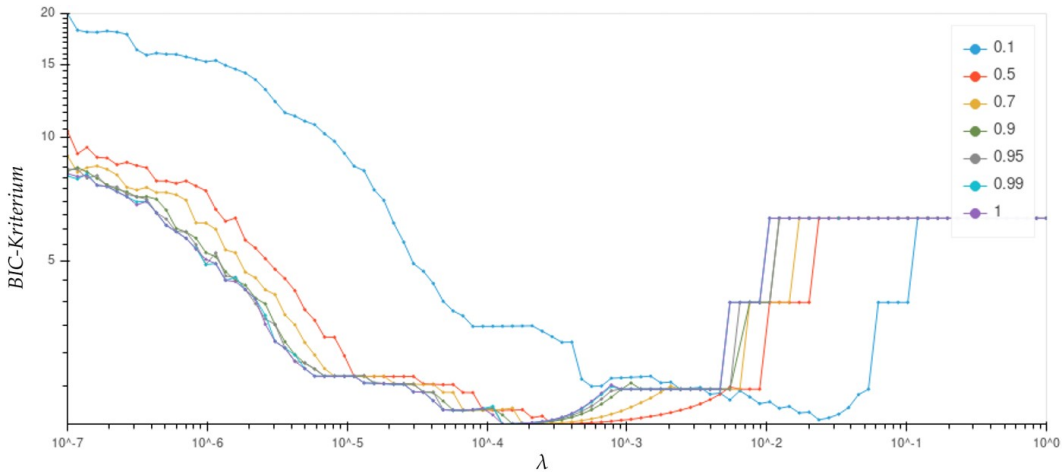
Im Zuge dieser Arbeit möchten wir nicht nur die Ergebnisse einiger Experimente, sondern auch das Verhalten des Algorithmus an sich beschreiben. Dafür sehen wir uns verschiedene Größen wie Laufzeit, Anzahl Iterationen und Toleranz an. Um ein Gefühl für die dünnbesetzte Hauptkomponentenanalyse zu bekommen, haben wir einen Überblick dieser Größen in Tabelle 6.1 erstellt. Interessant dabei ist vor allem die Abhängigkeit vom Hyperparameter λ. Je kleiner wir die Bestrafung wählen, desto länger dauert der Algorithmus und desto mehr Iterationen werden benötigt. Da bei kleinem λ mehr von Null verschiedene Einträge j in den Hauptachsen erlaubt werden, stimmen diese Beobachtung mit unseren Erwartungen überein. Der Anstieg der Zahl der Iterationen ist dadurch zu erklären, dass sich das Konvergenzkriterium auf alle Koeffizienten bezieht. Somit müssen bei einer Erhöhung von



(A) Anzahl freier Parameter in Abhängigkeit von λ.



(B) Erklärte Varianz des Datensatzes in Abhängigkeit von λ.



(C) Wert des BIC-Kriteriums in Abhängigkeit von λ.

ABBILDUNG 6.4: Die Abbildung zeigt wie sich eine Wahl der Hyperparameter α und λ mithilfe eines BIC-Kriteriums gestalten kann. Während λ auf der x-Achse aufgetragen ist, sind die unterschiedlichen Werte für α farblich markiert. Zu beachten ist die logarithmische Skala für λ.

λ	Laufzeit	Iterationen	Toleranz
10^{-6}	65235.76 sec	> 500	$2.271780 \cdot 10^{-1}$
10^{-5}	9158.24 sec	> 500	$4.132993 \cdot 10^{-4}$
10^{-4}	777.36 sec	99	$4.233382 \cdot 10^{-5}$
10^{-3}	201.49 sec	54	$5.531939 \cdot 10^{-5}$
10^{-2}	20.19 sec	5	0
10^{-1}	4.65 sec	1	0

TABELLE 6.1: Die Tabelle gibt einen Überblick über die Veränderung von Laufzeit, Iteration und Toleranz in Abhängigkeit von λ. Dabei wurde $\alpha = 0.5$ und die Anzahl an Hauptkomponenten $k = 10$ fest gewählt. Gerechnet wurde auf einem Intel Xeon Gold 6130F@2.10GHz.

ncalls	tottime	percall	cumtime	filename:lineno(function)
5010	4892.863	0.977	4896.461	_coordinate_descent.py:266(enet_path)
26075	4.948	0.000	4.948	{method 'reduce' of 'numpy.ufunc' objects}
6524	1.753	0.000	1.753	{method 'dot' of 'numpy.ndarray' objects}
76211	1.749	0.000	1.749	{built-in method numpy.array}
502	0.619	0.001	0.631	linalg.py:1458(svd)
5010	0.412	0.000	0.942	validation.py:800(check_random_state)

TABELLE 6.2: Die Tabelle zeigt die Ergebnisse eines Profilers für die eigene Implementierung mit $\lambda = 10^{-5}$ und $\alpha = 0.5$. Gerechnet wurde auf einem Intel Xeon Gold 6130F@2.10GHz.

j effektiv mehr Bedingungen erfüllt werden. Bezüglich der Laufzeit stimmen die Ergebnisse mit der Komplexität des Algorithmus von Abschnitt 5.3 überein. Diese wird dabei nicht überraschend von der Lösung der Elastic Nets dominiert. Hingegen sind die Kosten für das Minimieren über $\mathbf{A}$ unabhängig von den Hyperparametern und im Wesentlichen durch eine Singulärwertzerlegung bestimmt, welche für diesen Datensatz in einem Bruchteil einer Sekunde gelöst werden kann. Ein Blick auf Tabelle 6.2 zeigt die Ergebnisse eines Profilers und bestätigt unsere Behauptungen.

Logischerweise erhöht sich mit der Anzahl an Iterationen auch die Laufzeit des Algorithmus. Aus Tabelle 6.1 lässt sich aber nicht direkt erkennen, ob sich auch die Laufzeit pro Iteration mit λ verändert. In Abbildung 6.5 beobachten wir im Mittel eine Zunahme der Laufzeit pro Iteration bei Senkung von λ. Des Weiteren steigt die Laufzeit, wenn mehr Gewicht auf eine Lasso-Bestrafung gelegt wird, also α nahe 1 gewählt wird. Auch hier lässt sich das Verhalten auf die erhöhte Anzahl von Null verschiedener Einträge zurückführen.

Da wir eine maximale Anzahl von 500 Iterationen festgelegt haben, kommt es bei Werten $\lambda < 10^{-5}$ zu einer erhöhten Toleranz. Eine Erhöhung der Anzahl an Iterationen kann die Toleranz verringern, jedoch steigt damit auch die Laufzeit. Diese Obergrenze scheint für unsere Anwendung eine sinnvolle Wahl zu sein, muss aber für jeden Datensatz geeignet angepasst werden. Denkbar sind auch schwächere Konvergenzkriterien, die gegebenenfalls die Laufzeit verringern können.

6.4.4 Experimentelle Überprüfung der berechneten Varianzen

In Abschnitt 4.4 haben wir unterschiedliche Wege zur Berechnung der Hauptkomponenten und deren erklärte Varianz gezeigt. Um die Arbeit von Camacho et al. [Cam+20] experimentell zu überprüfen, werden wir vier Kriterien definieren, welche auf den unterschiedlichen

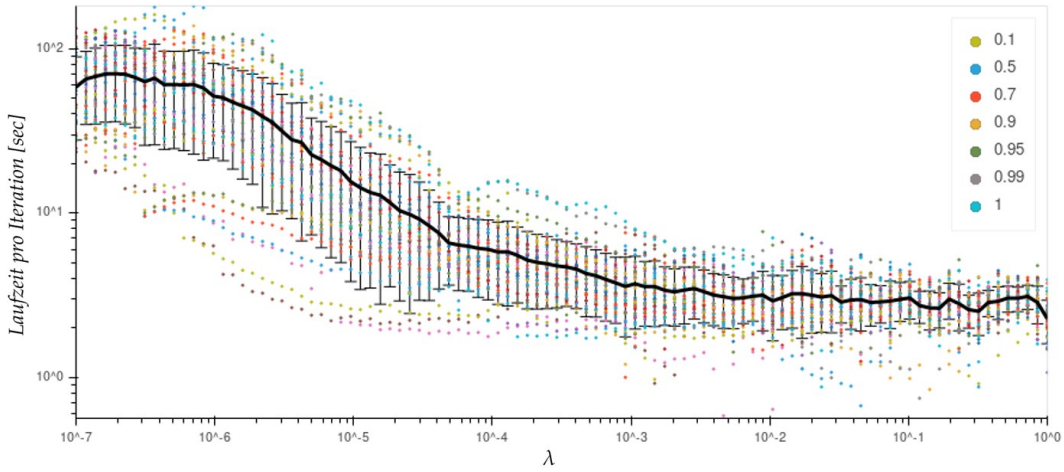


ABBILDUNG 6.5: In dieser Abbildung ist die Laufzeit pro Iteration bei Veränderung des Hyperparameters λ auf einer logarithmischen Skala zu sehen. Da auch α in unseren Experimenten variiert worden ist und mehrere Sensoren betrachtet werden, sehen wir mehrere Punkte je λ. Im Mittel klar zu erkennen ist ein Anstieg der Laufzeit bei Verringerung der Stärke der Bestrafung λ.

Vorgehensweisen basieren. Für jede dieser wird die Varianz der Residuen addiert und mit der Gesamtvarianz des Datensatzes normalisiert.

- TotQR: $\frac{\sum_{j=1}^{k} R_{jj}^2 + \operatorname{tr}\left(\widehat{\mathbf{E}}^\top \widehat{\mathbf{E}}\right)}{\operatorname{tr}\left(\mathbf{X}^\top \mathbf{X}\right)}$ (Vorgehensweies Zou et al.)
- TotZB: $\frac{\operatorname{tr}\left(\widehat{\mathbf{B}}\widehat{\mathbf{Z}}^\top \widehat{\mathbf{Z}}\widehat{\mathbf{B}}^\top\right) + \operatorname{tr}\left(\widehat{\mathbf{E}}^\top \widehat{\mathbf{E}}\right)}{\operatorname{tr}\left(\mathbf{X}^\top \mathbf{X}\right)}$ (Vorgehensweise Camacho et al.)

Bezüglich der Notation haben wir uns an Abschnitt 4.4 gehalten. Zwei weitere Kriterien TotQR* und TotZB* ergeben sich durch die Korrektur der Hauptkomponenten mit der Moore-Penrose-Inverse $\widehat{\mathbf{Z}}^* = \mathbf{X}\widehat{\mathbf{B}}^\top(\widehat{\mathbf{B}}^\top\widehat{\mathbf{B}})^+$. Falls alle Vorgehensweisen korrekt sind, können wir erwarten, dass jedes Kriterium den Wert 1 hat. In Abbildung 6.6 haben wir die Kriterien für unsere Experimente berechnet. Klar zu sehen ist, dass ohne eine Korrektur mit der Moore-Penrose-Inversen beide Varianten für die Varianzberechnung im Allgemeinem falsch sind. Auch wenn wir die Hauptkomponenten korrigieren, liefert die QR-Zerlegung keine richtigen Ergebnisse. Nur TotZB* hat in allen Fällen den Wert 1 und ist damit der einzig korrekte Weg, Hauptkomponenten und Varianzen zu berechnen. Die von Zou et al. [ZHT06] vorgeschlagenen Varianten sind also falsch und sollten nicht verwendet werden. Somit können wir die Erkenntnisse aus [Cam+20] experimentell bestätigen.

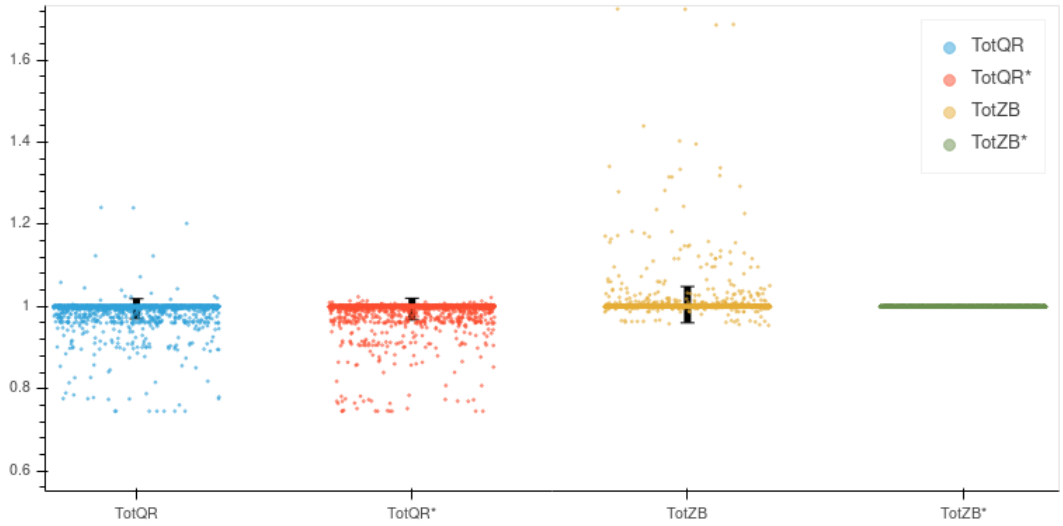


ABBILDUNG 6.6: Zu sehen sind die Ergebnisse der unterschiedlichen Vorgehensweise bei der Berechnung der Hauptkomponenten und der erklärten Varianz. Nur die von Camacho et al. vorgeschlagene Variante TotZB* errechnet diese korrekt. Jeder Punkt entspricht eines unserer Experimente aus Abschnitt 6.3.

Kapitel 7

Zusammenfassung

Im Zuge dieser Arbeit haben wir die Konstruktion, Theorie und Anwendung der dünnbesetzten Hauptkomponentenanalyse detailliert erläutert. Nach einer kurzen Einführung in die Thematik und deren Relevanz in der Praxis haben wir uns mit den mathematischen Grundlagen des Verfahrens beschäftigt. Besonders wichtig waren Kenntnisse aus der linearen Algebra und verallgemeinerter Regressionsmodelle. Anschließend wurde das wohl weitverbreiteste Dimensionsreduktionsverfahren, die Hauptkomponentenanalyse, in verschiedenen Formen vorgestellt. Grenzen, mögliche Erweiterungen und theoretische Aussagen des klassischen Verfahrens wurden beschrieben, um ein umfassendes Bild zu vermitteln. Im weiteren Verlauf wurde ein Kernproblem herausgestellt, was zur Idee der dünnbesetzten Hauptkomponentenanalyse führt. Zu diesem Zweck wurde ein in der Literatur vorgeschlagener Ansatz detailliert erklärt und eine numerische Lösung entwickelt. Mithilfe einer eigenen Implementierung haben wir die dünnbesetzte Hauptkomponentenanalyse auf einen realen Datensatz angewandt. Die beachtlichen Ergebnisse wurden kritisch hinterfragt und validiert. Darüber hinaus haben wir herausgearbeitet, wie das Verfahren in der Praxis einzusetzen ist.

7.1 Diskussion

Wir besitzen nun ein sehr gutes Verständnis über die Rolle der Dünnbesetzung und deren Einsatz für die Entwicklung transparenter Modelle. Allerdings gibt es noch immer Details und Zusammenhänge, welche es näher zu verstehen gilt. So ist man für die dünnbesetzte Hauptkomponentenanalyse noch immer auf der Suche nach Kriterien, welche eine automatisierte Wahl der Modellparameter ermöglichen. Teilweise mussten wir in unserer Anwendung manuelle Gewichtungen vornehmen, um brauchbare Ergebnisse zu erzielen. An dieser Stelle ist die Verwendung weiterer Kriterien aus der Informationstheorie denkbar. Vorerst sind wir aber dazu angehalten eigene Analysen durchzuführen und automatisierte Kriterien kritisch zu hinterfragen.

In Kapitel 4.2 haben wir erwähnt, dass es durchaus andere Wege gibt, das Problem der dünnbesetzten Hauptkomponentenanalyse zu relaxieren. Allgemein gibt es leider keinen besten Weg bzw. einen besten Algorithmus, so dass jeder Vorschlag seine Vor- und Nachteile beinhaltet. Bislang haben wir keine Kritik an dem von uns vorgestellten Ansatz durch Zou und Hastie [ZHT06] geäußert. Jedoch gibt es auch hier Schwierigkeiten, die bei der Verwendung zu beachten sind.

Im Vergleich zu anderen Methoden produziert der Algorithmus stärker korrelierte Hauptkomponenten. In der Praxis wäre es aber wünschenswert, wenn jede Hauptkomponente eine eigene Bedeutung im Kontext besitzt. Durch die Korrektur der Hauptkomponenten

mit der Moore-Penrose-Inversen, wie es in [Cam+20] vorgeschlagen wurde, wird eine Verringerung der Korrelation erreicht. Durch die Korrektur ist das Niveau der Korrelation mit anderen Methoden vergleichbar.

Weitere Kritik bezieht sich auf die Art der Modellierung. Das Sparse-PCA-Kriterium aus Kapitel 4 ist ein Optimierungsproblem über zwei Matrizen $\mathbf{A}$ und $\mathbf{B}$. Da sowohl $\mathbf{A}$ durch die Orthogonalitätsbedingung als auch $\mathbf{B}$ durch die verschiedenen Strafterme eingeschränkt ist, fehlt es dem Modell an Flexibilität. Je nach Stärke der Bestrafungen können wir daher nie die gesamte Varianz des Datensatzes erklären. Des Weiteren sind die dünnbesetzten Hauptachsen nicht wie üblich normalisiert. Zou und Hastie normalisieren die Hauptachsen daher am Ende des Algorithmus. Allerdings müsste die Matrix $\mathbf{A}$ dementsprechend angepasst werden, was in ihrer Arbeit nicht berücksichtigt wird und zu einem hohen Rekonstruktionsfehler führen kann. Daran erkennen wir, dass die Autoren primär an den dünnbesetzten Hauptachsen interessiert waren. Andere Verfahren erlauben eine flexiblere Modellierung, aber erreichen nicht immer eine vergleichbare Dünnbesetzung oder Laufzeit. Es fehlt der Literatur an einem weitreichendem Vergleich der verschiedenen Methoden in der Praxis.

7.2 Ausblick

In unserer Implementierung des Algorithmus haben wir bereits eine Verbesserung der Laufzeit gegenüber der von Zou und Hastie für hochdimensionale Datensätze erreicht. Wir können den Algorithmus weiter beschleunigen, indem wir einen Teil parallelisieren. Zwar müssen die Iterationen hintereinander ausgeführt werden, aber wir können das Elastic Net Subproblem für jede Hauptachse unabhängig voneinander berechnen. Mithilfe dieser Parallelisierung ist es möglich noch bessere Laufzeiten in der Praxis zu erhalten.

Auch wenn algorithmische Reduktionen in der theoretischen Informatik häufig genutzt werden, bleiben sie im Bereich des Machine Learnings oft unberücksichtigt. In unserem Fall ist es tatsächlich möglich, das Elastic Net auf ein anderes weitverbreitetes überwachtes Lernverfahren zurückzuführen, eine sog. *Support Vector Machine (SVM)*. Diese kommen sehr häufig für Mustererkennung bzw. Klassifikation der Beobachtungen in einem Datensatz zum Einsatz. In [Zho+15] wird gezeigt, dass zu jeder Elastic Net Instanz ein binäres Klassifikationsproblem konstruiert werden kann, so dass die trennende Hyperebene eine identische Lösung nach Skalierung liefert. Dies ermöglicht die Nutzung schneller parallelisierter CPU-basierten und auch GPU-basierten SVM-Lösern. Dadurch lässt sich in der Praxis eine um zwei Größenordnungen bessere Laufzeit erreichen. Für eine zukünftige Nutzung des von uns implementierten Algorithmus kann es daher sinnvoll sein, dieser Art Reduktion zu nutzen.

Zusammenfassend konnten wir in dieser Arbeit sowohl die Theorie der dünnbesetzten Hauptkomponentenanalyse aufarbeiten, als auch einen praktischen Teil beisteuern. Mithilfe der eigenen Implementierung haben wir beachtliche Ergebnisse erzielt, welche die Brauchbarkeit des Verfahrens für Frequenzdaten bestätigt.

Literatur

[ABB00] Orly Alter, Patrick O Brown und David Botstein. „Singular value decomposition for genome-wide expression data processing and modeling“. In: *Proceedings of the National Academy of Sciences* 97.18 (2000), S. 10101–10106. URL: https://doi.org/10.1073/pnas.97.18.10101.

[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.

[Bel61] Robert Bellman. „Curse of dimensionality“. In: *Adaptive control processes: a guided tour. Princeton, NJ* 3 (1961), S. 2.

[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.

[DJ94] David L Donoho und Jain M Johnstone. „Ideal spatial adaptation by wavelet shrinkage“. In: *biometrika* 81.3 (1994), S. 425–455. URL: https://doi.org/10.1093/biomet/81.3.425.

[DOM19] DOMO. *Data Never Sleeps 7.0*. Zugriff: 21. Februar 2020. 2019. URL: https://www.domo.com/learn/data-never-sleeps-7.

[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.

[FR13] Simon Foucart und Holger Rauhut. *A Mathematical Introduction to Compressive Sensing*. Bd. 1. Birkhäuser Basel, 2013.

[GD+04] John C Gower, Garmt B Dijksterhuis 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.

[Gir15] Christophe Giraud. *Introduction to high-dimensional statistics*. Boca Raton: CRC Press, Taylor & Francis Group, 2015. ISBN: 9781482237948.

[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.

[HBB96] Peter JB Hancock, A Mike Burton und Vicki Bruce. „Face processing: Human perception and principal components analysis“. In: *Memory & cognition* 24.1 (1996), S. 26–40. URL: https://doi.org/10.3758/BF03197270.

[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.

[IBM17] IBM. *How 10 industries are using big data to win big*. Zugriff: 21. Februar 2020. 2017. URL: https://www.ibm.com/blogs/watson/2016/07/10-industries-using-big-data-win-big/.

[Isr03] Adi Israel. *Generalized inverses: Theory and Applications*. New York: Springer, 2003. ISBN: 978-0-387-00293-4.

[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.

[JOB10] Rodolphe Jenatton, Guillaume Obozinski und Francis R. Bach. „Structured Sparse Principal Component Analysis“. In: *Artificial Intelligence and Statistics (AISTATS)* 9 (2010), S. 366–373. URL: https://arxiv.org/abs/0909.1440.

[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.

[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%2F978-0-387-39351-3.

[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 Ahipaşaoğlu, 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+04a] Robert Tibshirani u. a. *Diabetes Data*. Zugriff: 20. Januar 2020. 2004. URL: https://www4.stat.ncsu.edu/~boos/var.select/diabetes.html.

[Tib+04b] 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.

[Tse+88] Paul Tseng u. a. „Coordinate ascent for maximizing nondifferentiable concave functions“. In: (1988). URL: https://dspace.mit.edu/bitstream/handle/1721.1/3107/P-1840-19682973.pdf.

[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 ℓ_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.

[Zho+15] Quan Zhou u. a. „A reduction of the elastic net to support vector machines with an application to gpu computing“. In: *Twenty-Ninth AAAI Conference on Artificial Intelligence*. 2015. URL: https://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/viewPaper/9856.

[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.