Analyse dünnbesetzter Hauptachsen für Frequenzdaten

Tobias Bork

Geboren am 21. November 1997 in Reutlingen

6. November 2019

Bachelorarbeit Mathematik

Betreuer: Prof. Dr. Jochen Garcke

Zweitgutachter: Prof. Dr. X Y

Mathematisches Institut für Numerische Simulation

Mathematisch-Naturwissenschaftliche Fakultät der

Rheinischen Friedrich-Wilhelms-Universität Bonn

Danksagung

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

Inhaltsverzeichnis

Kapitel 1

Einführung

[2] [3] [6] [5] [7] [8] [15] [9] [11] [13] [16] [17] [18] [12]

1.1 Motivation

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

The goals of PCA are to

(1)

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

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

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

analyze the structure of the observations and the variables.

1.2 Dimensionsreduktionsverfahren

CURSE OF DIMENSIONALITY

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

Let’s say that you want to predict what the gross domestic product (GDP) of the United States will be for 2017. You have lots of information available: the U.S. GDP for the first quarter of 2017, the U.S. GDP for the entirety of 2016, 2015, and so on. You have any publicly-available economic indicator, like the unemployment rate, inflation rate, and so on. You have U.S. Census data from 2010 estimating how many Americans work in each industry and American Community Survey data updating

those estimates in between each census. You know how many members of the House and Senate belong to each political party. You could gather stock price data, the number of IPOs occurring in a year, and how many CEOs seem to be mounting a bid for public office. Despite being an overwhelming number of variables to consider, this just scratches the surface. TL;DR — you have a lot of variables to consider. If you've worked with a lot of variables before, you know this can present problems. Do you understand the relationships between each variable? Do you have so many variables that you are in danger of overfitting your model to your data or that you might be violating assumptions of whichever modeling tactic you're using? You might ask the question, "How do I take all of the variables I've collected and focus on only a few of them?" In technical terms, you want to "reduce the dimension of your feature space." By reducing the dimension of your feature space, you have fewer relationships between variables to consider and you are less likely to overfit your model. (Note: This doesn't immediately mean that overfitting, etc. are no longer concerns — but we're moving in the right direction!) Somewhat unsurprisingly, reducing the dimension of the feature space is called "dimensionality reduction." There are many ways to achieve dimensionality reduction, but most of these techniques fall into one of two classes: Feature Elimination Feature Extraction

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

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

1.3 Sparse Approximations / Representations

1.4 Interpretierbarkeit

1.5 Compressed Sensing Beispiel

Kapitel 2

Mathematische Grundlagen

2.1 Normen und deren Effekte

2.1.1 l0-Norm

2.1.2 l1-Norm

2.1.3 l2-Norm

2.2 Regression

Lineare Regression (Least Squares)

2.2.1 LASSO

2.2.2 Ridge Regression

2.3 Lineare Algebra

Ein Großteil der Hauptachsentransformation beruht auf Methoden der linearen Algebra. Daher werden wir im Folgendem die wichtigsten Begriffe einführen. Aufgrund des Anwedungsfalls werden wir uns hier auf reelle Vektorräume beschränken.

2.3.1 Orthogonalität

Definition 2.1 (Skalarprodukt **??**)**.** 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

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

linear. (Bilinearität)

(ii) $\langle x, y \rangle = \langle y, x \rangle$ für alle $x, y \in V$ (Symmetrie)

(iii) $\langle x, x \rangle \geq 0$ für alle $x \neq 0$ (Positive Definitheit)

Allgemein versteht man unter einem *euklidischer Vektorraum* ein Paar $(V, \langle \cdot, \cdot \rangle)$, welches aus einem reellem Vektorraum V und einem Skalarprodukt $\langle \cdot, \cdot \rangle$ auf V besteht.

Die durch das Skalarprodukt induziert Norm für $v \in V$ wird definiert durch:

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

Wir werden uns im Folgenden auf das *Standardskalarprodukt* im $\mathbb{R}^n$ beschränken. Dies ist gegeben durch

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

Die durch das Standardskalarprodukt induzierte Norm, ist die *euklidische Norm* oder l_2-Norm, welche wir bereits zuvor gesehen haben.

Definition 2.2 (Orthogonalität **??**)**.** Zwei Elemente v, w eines euklidischen Vektorraums 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 2.3 (Orthonormalbasis **??**)**.** Sei $\langle \cdot, \cdot \rangle : V \times V \longrightarrow \mathbb{R}$ ein Skalarprodukt. Ein System von Vektoren $(v_1, \ldots, v_n)$ in V wird als *Orthogonalbasis* (bzw. *Orthonormalbasis*) bezeichnet, wenn folgende Bedingungnen erfüllt sind:

(i) $(v_1, \ldots, v_n)$ ist eine Basis von V

(ii) $(v_1, \ldots, v_n)$ ist ein Orthogonalsystem (bzw. Orthonormalsystem)

Theorem 2.4 (Existenz einer Orthonormalbasis (Fischer Lineare Algebra))**.** *Jeder endlichdimensionale euklidische Vektorraum besitzt eine Orthonormalbasis.*

Theorem 2.5 (Verallgemeinerter Satz des Pythagoras **??**)**.** *Für orthogonale Vektoren u, v in einem euklidischem Vektorraum V gilt*

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

Der Begriff der Orthogonalität lässt sich auf lineare Abbildungen und somit auf Matrizen übertragen.

Theorem 2.6 (Orthogonale Abbildung **??**)**.** *Seien V, W euklidische Vektorräume. Eine lineare Abbildung $f : V \longrightarrow W$ heißt orthogonal oder isometrisch, wenn*

$$\langle f(v), f(w) \rangle = \langle v, w \rangle \quad \textit{für alle } v, w \in V$$

Definition 2.7 (Orthogonale Matrix **??**)**.** Eine Matrix $A \in \mathbb{M}(n \times n, \mathbb{R})$ heißt *orthogonal*, falls

$$\mathbf{A}^T \mathbf{A} = \mathbb{1}_n$$

Theorem 2.8 (Bosch)**.** *Für eine Matrix $\mathbf{A} \in \mathbb{M}(n \times n, \mathbb{R})$ sind die folgenden Bedingungen äquivalent:*

(i) $\mathbf{A}$ *ist orthogonal*

(ii) $\mathbf{A}^T\mathbf{A} = \mathbb{1}_n$

(iii) $\mathbf{A}\mathbf{A}^T = \mathbb{1}_n$

(iv) *Die Spalten von* $\mathbf{A}$ *bilden ein Orthonormalsystem*

(v) *Die Zeilen von* $\mathbf{A}$ *bilden ein Orthonormalsystem*

(vi) $\mathbf{A}$ *ist invertierbar und* $\mathbf{A}^{-1} = \mathbf{A}^T$

Von Wikipedia:

Definition 2.9 (Orthogonalprojektion (Wikipedia))**.** Eine *Orthogonalprojektion* auf einen Untervektorraum U eines Vektorraums 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 Lösung für die Orthogonalprojektion $P_U(v)$ herleiten.

Theorem 2.10 (??). *Ist* $(u_1, \dots, u_n)$ *eine Orthogonalbasis von* U*, so gilt für alle* $v \in V$

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

Ist $(w_1, \dots, w_n)$ *eine Orthonormalbasis von* U*, so gilt für alle* $v \in V$

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

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

$$P_U(v) = \sum_{i=1}^{n} \frac{v^T u_i}{u_i^T u_i} u_i = \sum_{i=1}^{n} \frac{u_i u_i^T}{u_i^T u_i} v = \mathbf{A}\mathbf{A}^T v$$

wobei die Spalten von $\mathbf{A}$ die normalisierten Vektoren der Orthogonalbasis sind, d.h. $\mathbf{A} = \left[\frac{u_1}{\|u_1\|} \middle| \cdots \middle| \frac{u_n}{\|u_n\|}\right]$.

Mithilfe von 2.5 lässt sich zeigen, dass der orthogonal auf den Unterraum projizierte Vektor den Abstand zwischen dem Ausgansvektor und dem Unterraum minimiert.

Theorem 2.11 (??). *Sei* U *ein Unterraum eines euklidischen Vektorraums* V*. Dann ist* $P_U(v)$ *die beste Näherung von* u *in* U*, d.h.*

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

2.3.2 Matrixzerlegungen

Eigenwerte, Eigenvektoren Singulärwerte

Diagonalisierbarkeit

Theorem 2.12 (Spektralsatz / Hauptachsentransformation (Beutelspacher)). *Jede symmetrische reelle Matrix ist diagonalisierbar und hat nur reelle Eigenwerte.*

Eigenwertzerlegung Singulärwertzerlegung

2.3.3 Matrixnorm und Rang (Eigenschaften?)

Definition 2.13 (Frobeniusnorm).

$$\|\mathbf{X}\|_F$$

Rang

Definition 2.14 (Rang). Eine Matrix hat Rang k ... wenn

Theorem 2.15 (Wörtlich von Wikipedia, Eckart-Young-Theorem). *The unstructured problem with fit measured by the Frobenius norm, i.e.,*

$$\textit{minimize} \quad \textit{over } \widehat{D} \quad \|D - \widehat{D}\|_F \quad \textit{subject to} \quad \operatorname{rank}\left(\widehat{D}\right) \leq r$$

has analytic solution in terms of the singular value decomposition of the data matrix. The result is referred to as the matrix approximation lemma or Eckart–Young–Mirsky theorem.[4] Let

$$D = U\Sigma V^\top \in \mathbb{R}^{m\times n}, \quad m \leq n$$

be the singular value decomposition of D and partition $U, \Sigma =: \operatorname{diag}(\sigma_1, \ldots, \sigma_m)$*, and V as follows:*

$$U =: \begin{bmatrix} U_1 & U_2 \end{bmatrix}, \quad \Sigma =: \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix}, \quad \textit{and} \quad V =: \begin{bmatrix} V_1 & V_2 \end{bmatrix},$$

where U_1 *is* $m \times r$*,* Σ_1 *is* $r \times r$*, and* V_1 *is* $n \times r$*. Then the rank-r matrix, obtained from the truncated singular value decomposition*

$$\widehat{D}^* = U_1\Sigma_1 V_1^\top,$$

is such that

$$\|D - \widehat{D}^*\|_F = \min_{\operatorname{rank}(\widehat{D})\leq r} \|D - \widehat{D}\|_F = \sqrt{\sigma_{r+1}^2 + \cdots + \sigma_m^2}.$$

The minimizer $\widehat{D}^*$ *is unique if and only if* $\sigma_{r+1} \neq \sigma_r$*.*

2.4 Signaltheorie

2.4.1 Fouriertransformation

2.4.2 Nyquist-Shannon Abtasttheorem

2.5 Statistik

Varianz, Erwartungswert

2.5.1 Empirische Kovarianzmatrix

2.6 Mannigfaltigkeit

2.7 Dictionary Learning

Kapitel 3

Hauptkomponentenanalyse

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

Die Hauptkomponentenanalyse ist ein weitverbreitetes multivariates statistisches Verfahren zur Dimensionsreduktion. Multivariate Verfahren zielen darauf ab, die in einem Datensatz enthaltene Zahl der Variablen zu verringern, ohne die darin enthaltene Information (zu verlieren) / (wesentlich zu reduzieren). Dadurch können umfangreiche Datensätze strukturiert, veranschaulicht und vereinfacht werden. Somit ist das Verfahren Teil der explorativen Statistik, welche Datensätze hinsichtlich ihrer Zusammenhänge analysiert. Die sich ergebende Struktur kann für weitere Analysezwecke ausgenutzt werden.

Aus diesem Grund hat die Hauptkomponentenanalyse in vielen Bereichen erfolgreich Anwendung gefunden. Darunter fällt die Erkennung handgeschriebener Zahlen, welche zum Beispiel zur automatischen Sortierung von Briefen nach Postleitzahl genutzt wird [6]. An diesem Beispiel lässt es sich besonders gut verdeutlichen, was es heißt Zusammenhänge zu analysieren und Strukturen auf den Daten zu finden. Man erhofft, dass nach Anwendung einer Dimensionsreduktion wie PCA auf den Datensatz 10 verschiedene Gruppierungen zu erkennen sind, die für die Ziffern 0 bis 9 stehen (siehe dazu Bild?). Optimalerweise gehören alle Datenpunkte im demselbem Cluster zur selben Ziffer. Außerdem korrespondieren nahe beieinanderliegende Cluster mit Ziffern, die ähnlich aussehen. Weitere Anwendung findet das Verfahren in der Bildverarbeitung. Hier kann es zum Beispiel zur Rauschunterdrückung [1] oder zur Gesichtserkennung [10] genutzt werden. Um Bilder für solch ein Verfahren nutzbar zu machen, werden einzelne Pixel oder patches, also lokale Gruppierungen von Pixeln, eines Bildes als Variable interpretiert.

Das mathematische Problem der Hauptkomponentenanalyse kann auf verschiedene Weisen beschrieben werden. Zunächst wollen wir es so konstruieren, dass die Idee des minimalen Informationsverlust im Vordergrund steht. Anschließend werden wir das Problem auf eine Singulärwertzerlegung zurückführen, die auch zur effizienten Implementierung genutzt wird. Des Weiteren werden wir die Hauptkomponentenanalyse als Regressionsproblem betrachten und die geometrische Interpretation weiter verdeutlichen. Zu Schluss werden wir einige theoretische Aussagen angeben, die für die folgenden Kapitel relevant sind.

3.1 Konstruktion

Gegeben sei ein Datensatz mit n samples 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. Denn eine Variable, die für alle Beobachtungen ähnliche Werte aufweist, ist nicht von Nutzen bei der Unterscheidung verschiedener samples. Um die Dimension zu reduzieren könnte man einfach nach den Eigenschaften größter Varianz suchen und alle Variablen unterhalb eines festgelegten Grenzwertes verwerfen. Dieses Vorgehen fällt allgemein unter die Methodik der *feature selection*. Die Hauptkomponentenanalyse verwendet allerdings ein anderes Prinzip, welches der Methodik der *feature extraction* zuzuordnen ist: Anstatt Eigenschaften mit hoher Varianz auszuwählen, konstruiert man neue Variablen, die die Bestehenden zusammenfassen. Variablen mit hoher Varianz werden in der Konstruktion einen größeren Beitrag spielen.

Konkret suchen wir also sukzessive nach einer Linearkombination der bestehenden Variablen. Finden wir nun zunächst die Richtung größter Varianz in unserem Datensatz, die erste *Hauptachse*. Der zugehörige Vektor spiegelt dabei den Beitrag bzw. den Informationsgehalt jeder einzelnen Variable wider. Anschließend finden wir weitere Hauptachsen, indem wir unter den Richtungen, die orthogonal zu allen vorherigen Hauptachsen sind, die mit der größten Varianz wählen. (Man iteriert diesen Prozess solange ...) Die Orthogonalität garantiert, dass die entstehenden Variablen unkorreliert sind. (Was hat das für einen Vorteil?) Nach der Identifizierung der Hauptachsen wollen wir unsere Beobachtungen bezüglich dieser darstellen. Wir erhalten die *Hauptkomponenten* unseres Datensatzes, indem wir die einzelnen samples auf die Hauptachsen transformieren. Aufgrund der schrittweisen Konstruktion verfügen Diese über eine sehr wichtige Sortierung. So beinhaltet die erste Hauptkomponente die meiste Information. Mit jeder weiteren Hauptkomponente erhält man mehr Information, aber der Informationsgewinn wird mit jeder Hauptkomponente geringer. Abbildung SCREE PLOT verdeutlicht diesen Verlauf.

Die eigentliche Dimensionsreduktion findet dann durch Selektion statt. Je nach Komplexität des Modells, welches man erreichen möchte, können so mehr oder weniger Hauptkomponenten ausgewählt werden. Je mehr Hauptkomponenten man auswählt, desto mehr Information erhält man über den Datensatz. Allerdings wird das Modell mit steigender Anzahl an Variablen immer komplizierter. Es gilt einen Punkt der Balance zu finden, der ein gutes Mittel aus Information und Komplexität liefert. Dieser kann vom Anwendungsfall abhängen. Wir werden uns mit diesem Thema weiter in 3.2 beschäftigen. Zusammenfassend haben wir somit unseren ursprünglichen Datensatz in neuen Hauptkomponenten konzentriert, die aber trotzdem einen Großteil an Information beinhalten.

Um dieses Prinzip zu veranschaulichen, wenden wir uns nun einem simplem Beispiel zu. Gegeben seien die Größe [cm] und das Gewicht [kg] zu 1000 Personen (Daten sind simuliert, keine real-world-data) (siehe dazu Abbildung). In diesem Fall ist also $n = 1000$ und $p = 2$. Bei Betrachtung der Abbildung fällt schnell auf, dass

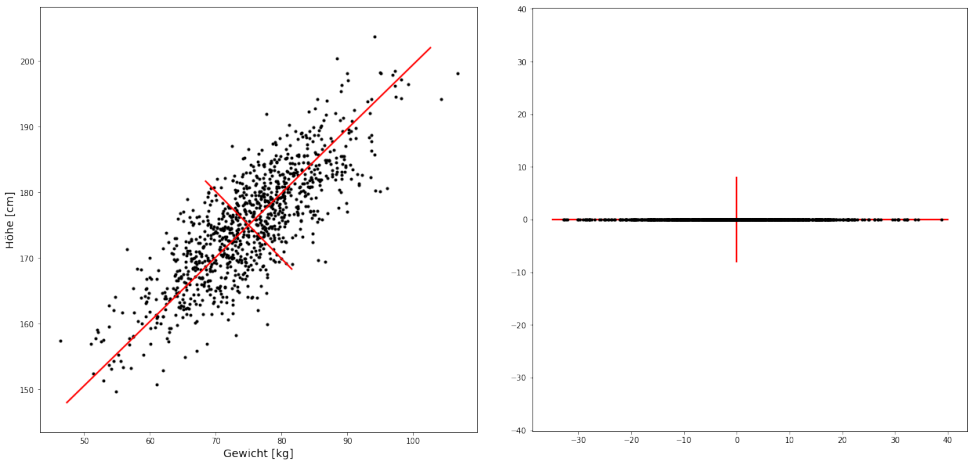

ABBILDUNG 3.1: Die Abbildung zeigt die Richtung größter Varianz
...

die beiden Variablen positiv korreliert sind, d.h. prinzipiell erkennt man folgende Tendenz: Je größer eine Person, desto schwerer ist sie.

Standardisierung

Bevor wir die Hauptkomponentenanalyse auf den Datensatz anwenden, gibt es aber noch einen wichtigen Bearbeitungsschritt zu beachten. Wenn eine Variable weniger variiert als eine Andere aufgrund der verwendeten Einheit oder Skala (meter oder kilo) kann dies zu ungewollten Ergebnissen führen. Ohne eine Vorbehandlung der Daten hat so im obigen Beispiel eine Änderung von 1m die gleiche Bedeutung wie eine Änderung von 1kg. (Satz schöner formulieren) Allerdings sind zwei Menschen, deren Größe 1m variiert, sehr verschieden, während zwei Menschen, die eine Differenz von 1kg haben, sehr ähnlich sind. Daher werden die Daten häufig einem sogenanntem preprocessing unterzogen. Ein zu diesem Zweck oft verwendetes Verfahren ist die Standardisierung oder auch z-Transformation genannt. In diesem Schritt werden die Variablen so transformiert, dass sie *vergleichbarer* werden. Seien dazu Y_i die Zufallsvariablen mit Erwartungswert $E[Y_i] = \mu$ und Varianz $Var[Y_i] = \sigma^2$. So erhält man die zugehörigen standardisierten Zufallsvariablen X_i durch Zentrierung und anschließender Division durch die Standardabweichung $X_i = \frac{Y_i - \mu}{\sigma}$. Somit gilt dann:

- $E[X_i] = 0$ für alle $1 \leq i \leq p$
- $Var[X_i] = 1$ für alle $1 \leq i \leq p$

Mathematisch gesehen wendet man das Verfahren also nicht auf die Kovarianzmatrix, sondern auf die Korrelationsmatrix an.

3.1.1 Problemformulierung als Varianzmaximierung

Wir wollen nun die Intuition des minimalen Informationsverlust mathematisch beschreiben. Gegeben sei dazu eine Matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$, wobei n die Anzahl der Samples bzw. Beobachtungen und p die Anzahl der Variablen ist. Wir nehmen im Folgenden ohne Beschränkung der Allgemeinheit an, dass die Variablen zuvor zentriert wurden. Aufgabe der Hauptkomponentenanalyse ist es nun sukzessive Richtungen größter Varianz zu finden. Die erste Hauptkomponente ist definiert durch $Z_1 = \sum_{j=1}^{p} v_{1j} X_j = \mathbf{X}v$ wobei die Hauptachse $v_1 = (v_{11}, \ldots, v_{1p})^T$ so gewählt wird, dass die Varianz von Z_1 maximiert wird, d.h.

$$v_1 = \underset{\|v\|_2=1}{\arg\max} \operatorname{Var}[\mathbf{X}v] = \underset{\|v\|_2=1}{\arg\max}\, v^T \mathbf{K}_{xx} v$$

mit $\mathbf{K_{xx}} = \frac{\mathbf{X}^T\mathbf{X}}{n-1}$ als Stichprobenkovarianzmatrix. Die restlichen Hauptachsen können nun sukzessive definiert werden.

$$v_{k+1} = \underset{\|v\|=1}{\arg\max}\, v^T \mathbf{K}_{xx} v$$

$$v_{k+1}^T v_l = 0 \quad \forall 1 \leq l \leq k$$

Man sucht also unter den Richtungen, die orthogonal zu allen bisherigen Hauptachsen sind, diejenige, die die Varianz maximiert. Wie oben beschrieben erhält man dann die Hauptkomponenten, also die Darstellung der Daten bezüglich der neu gefundenen Hauptachsen, durch Projektion der Daten $Z_i = \mathbf{X}v_i$. [18] CITE JOLLIFE

Wie wir bereits in THEOREM gesehen haben, entsprechen die Eigenvektoren der Kovarianzmatrix genau den Richtungen maximaler Varianz. Daher können wir anstatt sukzessiver Berechnung einzelner Hauptachsen die Kovarianzmatrix $\mathbf{K}_{xx}$ direkt diagonalisieren. Dies ist möglich, da $\mathbf{K}_{xx}$ symmetrisch ist. Die Diagonalisierung ergibt

$$\mathbf{K}_{xx} = \mathbf{V}\mathbf{L}\mathbf{V}^T$$

wobei $\mathbf{L}$ eine Diagonalmatrix mit Eigenwerten λ_i und $\mathbf{V}$ die Matrix der Eigenvektoren ist, d.h. jede Spalte entspricht einem Eigenvektor von $\mathbf{K}_{xx}$. Somit können die Hauptachsen direkt aus $\mathbf{V}$ abgelesen werden. Die Projektion der Daten auf die Hauptachsen wird dann wie zuvor durch Multiplikation der Beobachtungen mit den Eigenvektoren erreicht.

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

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

3.1.2 Formulierung als Singulärwertzerlegung

Es gibt einen engen Zusammenhang zwischen der Diagonalisierung der Kovarianzmatrix $\mathbf{K}_{xx} = \mathbf{X}^T\mathbf{X}$ und der Singulärwertzerlegung von $\mathbf{X}$. Diese Beziehung können wir nutzen, um das Problem neu zu formulieren. Eine Singulärwertzerlegung der Matrix $\mathbf{X}$ ergibt

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

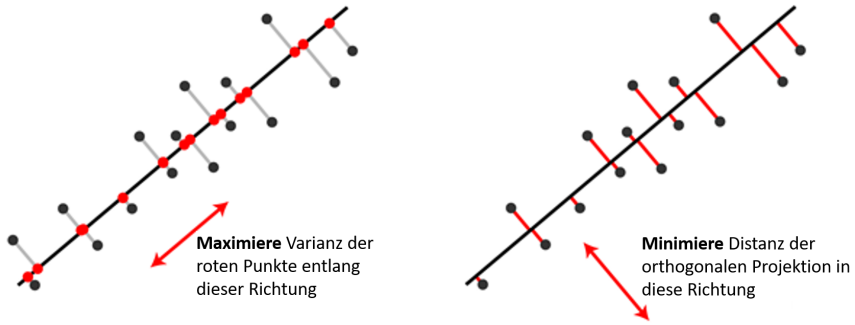

ABBILDUNG 3.2: Die Abbildung zeigt die Äquivalenz von Maximierung der Varianz und Minimierung der Distanz der orthogonalen Projektion

wobei $\mathbf{D}$ eine Diagonalmatrix mit Singulärwerten $d_1, \ldots, d_p$, $\mathbf{U}$ eine orthogonale $n \times p$ und $\mathbf{V}$ eine orthogonale $p \times p$ Matrix ist. Nun sieht man aufgrund der Orthogonalität von $\mathbf{U}$, dass

$$\mathbf{K}_{xx} = \mathbf{X}^T\mathbf{X} = \mathbf{V}\mathbf{D}\mathbf{U}^T\mathbf{U}\mathbf{D}\mathbf{V}^T = \mathbf{V}\mathbf{D}^2\mathbf{V}^T$$

Wegen der Eindeutigkeit der Diagonalisierung(stimmt das?) ist $\mathbf{V}$ nun wie zuvor die Matrix der Eigenvektoren und somit der Hauptachsen. Ebenso stehen die Singulärwerte durch

$$\lambda_i = \frac{d_i^2}{n-1}$$

in Beziehung mit den Eigenwerten der Kovarianzmatrix. Die Hauptkomponenten kann man somit auch durch $\mathbf{XV} = \mathbf{UD}$ erhalten.

Computing PCA using Eigen value decomposition of the sample covariance matrix: We first have to compute the covariance matrix, which is $O(p^2n)$ and then compute its eigenvalue decomposition which is $O(p^3)$ giving a total cost of $O(p^2n + p^3)$ (https://arxiv.org/pdf/1503.05214.pdf)

Computing PCA using SVD of the data matrix: Svd has a computational cost of $O(p^2n)$

Numerical Stability? Which method is preferrable in the $n << p$ case?

3.1.3 Formulierung als beste Rang k Rekonstruktion

Further multiplying the first k PCs by the corresponding principal axes $\mathbf{V}^T k$ yields $\mathbf{X}_k = \mathbf{U}_k\mathbf{S}_k\mathbf{V}_k^T$ matrix that has the original $n \times p$ size but is of lower rank (of rank k). This matrix $\mathbf{X}_k$ provides a reconstruction of the original data from the first k PCs. It has the lowest possible reconstruction error

3.1.4 Formulierung als Regressionsproblem

Wir widmen uns nun einer letzten Formulierung der Hauptkomponentenanalyse, die eine geometrische Interpretation ermöglicht. Hierbei versucht man einen k-dimensionalen $(k < n)$ Unterraum zu finden, der die Daten bestmöglich approximiert. Wir werden diese Problemstellung nun mathematisch formulieren.

Sei dazu x_i die i-te Beobachtung, also die i-te Zeile von $\mathbf{X}$ und $\mathbf{V}_k = [V_1|\cdots|V_k]$ eine $p \times k$ orthonormale Matrix. Nun projizieren wir jede Beobachtung orthogonal auf den durch $V_1, \ldots V_k$ aufgespannten Unterraum. Die orthogonale Projektion wird wie in REF beschrieben durch Multiplikation mit dem Operator $\mathbf{V}_k\mathbf{V}_k^T$ erreicht. Die auf den linearen Unterraum projizierten Daten ergeben sich also durch $\mathbf{V}_k\mathbf{V}_k^T x_i$. Um die Daten bestmöglich in diesem niedrigdimensionalen Raum darzustellen minimiert man nun die Distanz zwischen jeder Beobachtung und seiner Projektion. Ein Weg, um die beste Projektion zu definieren is l_2 Approximation erhält man folgendes Problem [17]: (Hier auch noch schreiben warum man den zweiten Term braucht, Eindeutigkeit von PCA)

$$\hat{\mathbf{V}}_k = \arg\min_{\mathbf{V}_k} \sum_{i=1}^{n} \left\| x_i - \mathbf{V}_k\mathbf{V}_k^T x_i \right\|^2 + \lambda \sum_{j=1}^{k} \|\beta_j\|^2$$

$$\mathbf{V}_k^T\mathbf{V}_k = I_{k\times k}$$

Man kann zeigen, dass die Lösung dieses Problems genau den ersten k Hauptachsen entspricht. Wir haben dies in einem Theorem 3.3 festgehalten. [14] Zum besseren Verständnis hilft 3.2, welches die Äquivalenz von Maximierung der Varianz und Minimierung der orthogonalen Projektion verdeutlichen soll. Jeder Datenpunkt ist hier in 2 Dimensionen dargestellt. Versucht man nun die Daten bestmöglich auf einen 1-dimensionalen Unterraum, also eine Linie, orthogonal zu projizieren erhält man denselben Vektor, den man aus Sicht der Varianzmaximierung auch erhalten hätte.

Aus dieser Interpretation leitet sich auch der Name des linearen Dimensionreduktionsverfahrens ab, denn die Daten werden auf den niedrigdimensionaleren Raum linear transformiert. Ausgehend von dieser Formulierung als Regressionsproblem werden wir im nächsten Kapitel die Variante der dünnbesetzten Hauptkomponentenanalyse beschreiben.

3.2 Selektion der Hauptkomponenten

Wie viele Hauptkomponenten sollen wir auswählen?

A simple approach is to choose the number of PCs for the variance to achieve a predetermined percentage. say 95Most existing approaches to determining the number of PC's use an index that is monotonlcally decreasing. The number of PC's is chosen when there is no significant decrease in the index after adding a PC. These approaches based on monotonic indices are subjective because (i) there may be a rather constant decrement in the index; and (ii) there can be more than one location which satisfies the criterion

Abbildung Scree Plot

Optimal singular threshold [4]

3.3 Grenzen der Anwendbarkeit

Obwohl die Hauptkomponentenanalysen 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 vorest hinsichtlich folgender Gesichtspunkte zu untersuchen:

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

Wie in REF beschrieben versuchen wir Daten in einen niedrigdimensionaleren linearen oder affinen Unterraum zu transformieren. Es kann aber durchaus vorkommen, dass es keine lineare Beziehung zwischen den Variablen gibt. Nichtlineare Strukturen können von PCA nicht erfasst werden und gehen somit verloren. [14] Vidal et al. zeigen diese Grenze konkret am Beispiel von Porträt-Fotos auf. Seit der Entstehung von PCA gab es aber zahlreiche nicht-lineare Erweiterungen. So nutzt zum Beispiel Kernel PCA den *Kernel Trick* aus, bei welchem man die Daten zuerst durch eine nichtlineare Transformation in ein höherdimensionalen Raum einbettet von dem man sich erhofft, dass die Daten in diesem linear verteilt. Erst anschließend wird dann die eigentliche Reduktion durchgeführt. Hierbei muss man die Daten aber nicht im höherdimensionalen Raum auswerten. CITE. Andere Erweiterungen, die allgemein unter *manifold learning* zusammengefasst werden können, basieren auf der Idee, dass die Dimension des Datensatz nur künstlich hoch ist. Man versucht die lokale Geometrie der Mannigfaltigkeit (Begriff erklären?) zu approximieren und damit direkt eine niedrigdimensionale Einbettung zu erhalten. Hierunter fallen zum Beispiel die multidimensionale Skalierung oder ISOMAP.

Damit der Datensatz für eine Dimensionsreduktion per PCA geeignet ist, müssen die verschiedenen Variablen einen gewissen Grad an Korrelation aufweisen. Im extremen Fall der Unabhängigkeit der Variablen bewirkt eine Hauptachsentransformation nichts. Reduziert man dann die Anzahl der Hauptkomponenten verliert man mit jeder Variable einen Großteil der Information.

Ein weiterer Gesichtspunkt ist die Vollständigkeit eines Datensatzes. Finden wir fehlende oder korrupte Einträge in unserem Datensatz vor, kann die klassische Hauptkomponentenanalyse ... . Für dieser Art Probleme existieren entsprechende Ergänzungen von PCA wie zum Beispiel in cite und cite. Ausreißer in den Daten können die Resultate drastisch beeinflussen. Genaue Effekte überlegen und CITE. Aus diesem Grund sollten Ausreißer vor der Anwendung von PCA entfernt werden.

Außreiser in den Daten.

Anzahl der Variablen zu hoch.

Darüber hinaus gibt es noch eine Reihe Spezialfälle, bei denen Probleme auftreten können. So kann es zum Beispiel passieren, dass die relevanten Informationen in den Variablen mit niedriger Varianz versteckt sind. Da die Hauptkomponentenanalyse gerade diese Variablen vernachlässigt, wird sich unter Umständen nicht die

Loss Functions	regularizer	constraints
quadratic	L2 norm	Nonnegative
(real data)	(small factors)	(additive factors)
absolute	L1 norm	
(robust to outliers)	(sparse factors)	
logistic	Derivative penalties	
(binary data)	(smooth factors)	
Poisson		
(integer data)		
circular		
(angular data)		

TABELLE 3.1: Allgemeines Schema zu PCA Erweiterungen

erwünschte Struktur auf den Daten ergeben. Es bedarf anderer Methoden mit anderen Ansätzen, um eine Dimensionsreduktion zu ermöglichen. Oftmals weiß man aber im Vorhinein nicht, in welchen Variablen diese Unterscheidungsmöglichkeit versteckt ist.

Das wohl wichtigste/größte Hindernis im Zuge dieser Arbeit ist sicherlich die durch die Transformation entstehenden Interpretationsschwierigkeiten. Jede Hauptkomponente entsteht wie oben beschrieben durch eine Linearkombination der Ausgangsvariablen. Während die Ausgangsvariablen Bedeutungen wie Gewicht oder Größe hatten ist in vor allem in hochdimensionalen Fällen eine Interpretation der Hauptkomponenten nur schwer möglich (Rotation Techniques CITE). Dieser Interpretationsverlust ist Ausgangspunkt der Idee der dünnbesetzten Hauptkomponentenanalyse, genannt sparse PCA. Diesem Verfahren ist das folgende Kapitel gewidmet.

3.4 Erweiterungen der Hauptkomponentenanalyse

Wie wir bereits gesehen haben, gibt es viele verschiedene Erweiterungen von PCA. Die meisten kann man unter folgendem Schema zusammenfassen: (Welche genau?)

$$\min_{\mathbf{U},\mathbf{V}} \underbrace{\left\|\mathbf{X}-\mathbf{U}\mathbf{V}^T\right\|_F}_{\text{Loss Function}} + \underbrace{\lambda_u f_u(\mathbf{U}) + \lambda_v f_v(\mathbf{V})}_{\text{Regularisierung}}$$
$$\text{subject to} \quad \underbrace{\mathbf{U} \in \Omega_u, \mathbf{V} \in \Omega_v}_{\text{Nebenbedingungen}}$$

3.5 Implementierung

Allgemein ist dies kein konvexes Problem, aber bikonvex, also in jeder Komponente. Somit ergibt sich der einfache folgende Algorithmus

3.6 Theoretische Aussagen

non convex problem that can be solved efficiently by truncated SVD.

Algorithm 1 Alternating minimization

procedure ALTERNATE(U, V)
  choose initial starting Points $\mathbf{W}^{(0)}$ and $\mathbf{C}^{(0)}$
  $n \leftarrow 0$
  while not converged **do** ▷ Definiere Abbruchkriterium
    $\mathbf{W}^{(n+1)} \leftarrow$ minimize over $\mathbf{W}$ while holding $\mathbf{C} = \mathbf{C}^{(n)}$ constant.
    $\mathbf{C}^{(n+1)} \leftarrow$ minimize over $\mathbf{C}$ while holding $\mathbf{W} = \mathbf{W}^{(n+1)}$ constant.
    $n \leftarrow n + 1$
  end while
end procedure

Baldi Hornik 1989 all local minima are solutions to pca all non optimal critical points are saddle points or maxima

Theorem 3.1. *PCA always gives unique solution.*

Theorem 3.2 ([14]). *Sei* $\mathbf{X} \in \mathbb{R}^n$ *und* $\mathbf{A}_{p,k} = [\alpha_1, \dots \alpha_k]$

Theorem 3.3. *PCA inconsistent for* $n \ll p$.

Kapitel 4

Dünnbesetzte Hauptkomponentenanalyse

Ein Nachteil der Hauptkomponentenanalyse ist, dass sich die neuen Variablen meist aus einer Linearkombination aller bestehenden Variablen zusammensetzt. Dies macht es besonders für hochdimensionale Daten schwierig die Hauptachsen zu interpretieren. Oft können somit nicht die relevanten features/Variablen herausgelesen werden. Es kann durchaus passieren, dass nicht alle Variablen relevant zur Strukturerkennung sind.

4.1 Motivation

4.2 Problemformulierung

NP-schwere Formulierung

4.3 Relaxation / Approximation Ideen

4.4 Konstruktion

Sparse PCA Kriterium.

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

subject to $\mathbf{A}^T\mathbf{A} = I_{k\times k}$

4.5 Theoretische Aussagen Sparse PCA

z.B. wie werden neue Varianzen berechnet

Kapitel 5

Implementierung

5.1 Implementierung nach original paper

Eigene Implementierung

5.2 Implementierung in scikit-learn in python

5.3 Laufzeitvergleich

Kapitel 6

Anwendung

6.1 Anwendung auf Simulationsdaten

Vergleich Tabelle PCA / Sparse PCA (Loadings)

6.2 Der Datensatz

6.3 Anwedung auf Frequenzdaten

6.4 Auswertung der Ergebnisse

6.5 Vergleich mit PCA Resultaten

6.6 Hyperparameter

Veränderung des Hyperparameters und dessen Effekte

6.6.1 Zeit

6.6.2 Effekt auf Resultate

Kapitel 7

Ausblick / Zusammenfassung

7.1 Einsetzbarkeit

Wann ist die Methode sinnvoll einzusetzen?

7.2 Übertragbarkeit

Übertragbarkeit auf andere Datensätze

7.3 Ongoing Research / Weitere Techniken

Literatur

[1] 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.

[2] Michael Elad. *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. Bd. 1. Springer Science+Business Media, 2010.

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

[4] 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.

[5] Rémi Gribonval, Rodolphe J Jenatton und Francis R. Bach. „Sparse and spurious: dictionary learning with noise and outliers“. In: *IEEE Transactions on Information Theory* (2014). URL: http://arxiv.org/abs/1407.5155.

[6] Trevor Hastie, Robert Tibshirani und Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Bd. 2. Springer-Verlag New York, 2009.

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

[8] 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.

[9] Julien Mairal u. a. „Online Dictionary Learning for Sparse Coding“. In: *Proceedings of the 26th Annual International Conference on Machine Learning*. ICML '09. ACM, 2009, S. 689–696. URL: http://doi.acm.org/10.1145/1553374.1553463.

[10] 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.

[11] 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.

[12] 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.

[13] Ryan J. Tibshirani. „The Lasso Problem and Uniqueness“. In: (2012). URL: https://arxiv.org/abs/1206.0313.

[14] 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.

[15] Kazuyoshi Yata und Makoto Aoshima. „Effective PCA for high-dimension, low-sample-size data with noise reduction via geometric representations“. In:

Journal of Multivariate Analysis 105.1 (2012), S. 193–215. URL: https://doi.org/10.1016/j.jmva.2011.09.002.

[16] 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.

[17] 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.

[18] Hui Zou und Lingzhou Xue. „A Selective Overview of Sparse Principal Component Analysis“. In: *Proceedings of the IEEE* 106.8 (2018), S. 1311–1320. URL: https://ieeexplore.ieee.org/document/8412518.