

Bettina Laure, BSc

Geometrische Approximationsmethoden und ihre Anwendung in der Qualitätssicherung

MASTERARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieurin

Masterstudium Mathematische Computerwissenschaften

eingereicht an der

Technischen Universität Graz

Betreuer

Univ.-Prof. Mag.rer.nat. Dipl.-Ing. Dr.techn. Johannes Wallner

Institut für Geometrie

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Das in TUGRAZonline hochgeladene Textdokument ist mit der vorliegenden Masterarbeit identisch.

Datum

Unterschrift

Kurzfassung

In dieser Arbeit beschäftigen wir uns mit dem Lösen von Approximationsproblemen im Sinne der minimalen orthogonalen Distanz. Es werden verschiedene Methoden vorgestellt, wie sich Punktmengen so mit geometrischen Objekten approximieren lassen, dass die Quadratsumme der orthogonalen Abstände der einzelnen Punkte zum Objekt minimal wird.

Das Approximieren von Punktmengen mittels Geraden und Ebenen mit Hilfe der Hauptkomponentenanalyse bildet den Einstieg in die vorgestellten Lösungsmethoden. Im Weiteren werden zwei robuste Methoden der Hauptkomponentenanalyse vorgestellt, das Zentrieren der Daten mittels $\|\cdot\|_1$ -Norm und eine iterative Form der Hauptkomponentenanalyse mit gewichteten Distanzen.

Anschließend werden weitere iterative Lösungsverfahren für die Methode der kleinsten Quadrate behandelt, zu denen das Gauss-Newton-Verfahren und die Levenberg-Marquardt-Methode zählt. Um diese nichtlinearen Methoden anwenden zu können, wird in einem kurzen Exkurs auf Lösungsmethoden zum Lösen von linearen Gleichungssystemen eingegangen.

Ein Beispiel aus dem Bereich der Messtechnik zeigt schließlich die Anwendung der zuvor vorgestellten Theorie. Hier wird ein durch eine Punktmenge dargestelltes Profil eines Werkzeuges mittels mehrerer geometrischer Objekte mit Hilfe der Levenberg-Marquardt-Methode approximiert.

Abstract

This thesis focuses on methods for the solution of approximation problems with respect to the minimal orthogonal distance. Various methods for approximating point sets with geometric objects are presented, that are minimizing the sum of squared orthogonal distances of the given points to the object.

Principal component analysis is the first method presented in this thesis and is used for approximating point sets with lines and planes. Two robust approaches on the principal component analysis show how to deal with outliers in point sets. The first one is using the $\|\cdot\|_1$ -norm for centering the data and the second approach is an iterative form of the principal component analysis that is using weighted distances.

Both the Gauss-Newton method and the Levenberg-Marquardt method are two iterative procedures for solving least-squares problems. To use these non-linear procedures some methods for solving a system of linear equations will be presented.

An example from the field of metrology is used to show the application of the Levenberg-Marquardt method. Given a point set in the form of a profile of a cutting edge, the algorithm is used to iteratively approximate the set with some geometric objects.

Inhaltsverzeichnis

| | |
|--|------------|
| Kurzfassung | iii |
| Abstract | iv |
| 1 Einleitung | 1 |
| 2 Definitionen | 3 |
| 2.1 Lineare Algebra | 5 |
| 2.2 Statistik | 8 |
| 3 Funktionelles und geometrisches Fitting | 12 |
| 4 Approximation mit einfachen Objekten | 17 |
| 4.1 Geometrische Approximation mit Hilfe einer Geraden im \mathbb{R}^3 | 17 |
| 4.2 Hauptkomponentenanalyse | 19 |
| 4.2.1 PCA zum Approximieren mit einfachen Objekten | 23 |
| 4.2.2 PCA und Singulärwertzerlegung | 31 |
| 4.2.3 Robuste Hauptkomponentenanalyse | 32 |
| 5 Iterative Lösungsverfahren für die Methode der kleinsten Quadrate | 40 |
| 5.1 Die Newton-Methode | 42 |
| 5.2 Das Gauss-Newton-Verfahren | 45 |
| 5.3 Das Levenberg-Marquardt-Verfahren | 46 |
| 5.4 Lösungsmethoden für lineare Gleichungssysteme | 50 |
| 5.4.1 Die Cholesky-Zerlegung | 50 |
| 5.4.2 Die QR-Zerlegung | 52 |
| 5.4.3 Die Singulärwertzerlegung | 53 |
| 5.4.4 Vergleich der drei Zerlegungsmethoden | 55 |
| 5.5 Approximation einer Punktmenge mit einem Kreis im \mathbb{R}^2 | 55 |
| 6 Ein Anwendungsbeispiel | 59 |
| 7 Fazit und Ausblick | 69 |
| Literaturverzeichnis | 71 |

Abbildungsverzeichnis

| | | |
|-----|--|----|
| 2.1 | Darstellungen eines Kreises | 4 |
| 3.1 | Unterschied zwischen Interpolation (links) und Approximation (rechts) | 13 |
| 3.2 | Unterschied zwischen funktioneller (links) und geometrischer (rechts) Approximation | 15 |
| 3.3 | Approximation einer Punktmenge mit einer Geraden mittels funktioneller Distanz in X (grün), in Y (orange) und mittels geometrischer Distanz (blau) im \mathbb{R}^2 | 15 |
| 4.1 | Ergebnis der Approximation mittels einer Geraden im \mathbb{R}^2 | 20 |
| 4.2 | Die zwei Hauptkomponenten einer Datenmenge im \mathbb{R}^2 | 22 |
| 4.3 | Approximieren einer Datenmenge mittels einer Ebene im \mathbb{R}^3 | 30 |
| 4.4 | Ergebnis der Approximation mittels einer Geraden mit und ohne Zentrieren der Punktmenge | 33 |
| 4.5 | Approximieren von Datenpunkten ohne (links) und mit Ausreißern (rechts) mittels PCA | 34 |
| 4.6 | Gewichtsfunktion $y = \frac{1}{1+(\frac{x}{4})^\beta}$ für verschiedene Werte für β | 35 |
| 4.7 | PCA mit Gewichtsiteration zum Approximieren einer Geraden in Datenpunkte mit Ausreißern | 37 |
| 4.8 | Approximieren von Datenpunkten mit Ausreißern mittels Zentrierung mit Hilfe des geometrischen Medians | 39 |
| 5.1 | Graphische Darstellung der univariaten Newton-Methode | 43 |
| 5.2 | Gauss-Newton-Algorithmus | 47 |
| 5.3 | Levenberg-Marquardt-Algorithmus | 49 |
| 5.4 | Iterationsschritte des Kreisfits | 57 |
| 6.1 | Einteilung eines Profils in vier Abschnitte | 60 |
| 6.2 | Ein unrealistisches Ergebnis der Approximation | 62 |
| 6.3 | Zuordnung der Punkte für die obere rechte Seite des Profils | 62 |
| 6.4 | Zuordnung eines Punktes | 63 |
| 6.5 | Zuordnung der Punkte für die untere Seite des Profils | 64 |
| 6.6 | Zuordnung der Punkte für die obere linke Seite des Profils | 64 |
| 6.7 | Ergebnis der Approximation für Kante 1 | 67 |
| 6.8 | Ergebnis der Approximation für Kante 2 | 68 |
| 6.9 | Ergebnis der Approximation für Kante 3 | 68 |

1 Einleitung

Experimentelle Daten mittels geometrischen Objekten anzunähern, ist eine der Hauptaufgaben im Bereich der Mustererkennung und Computergrafik. Die Darstellung solcher Daten mittels Objekten, wie Strecken, Kreisbögen und Kurven, wird in vielen technischen Bereichen verwendet, um Daten oder Objekte anhand ihrer geometrischen Struktur bewerten zu können. Werden gegebene geometrische Objekte (zum Beispiel Punktwolken) mit Hilfe von weiteren geometrischen Objekten beschrieben, spricht man von einer Approximation.

Die Gebiete der Approximationstheorie sind weit gefächert und viele berühmte Mathematiker haben zu ihrer Entwicklung beigetragen.

Diese Arbeit widmet sich speziellen Approximationsproblemen, die besonders im Bereich der Qualitätssicherung oder der Bildverarbeitung Anwendung finden. Es werden Lösungsmethoden vorgestellt, die eine gegebene Punktmenge mittels geometrischer Objekte im Sinne der minimalen orthogonalen Distanz approximieren.

Allgemein versucht man bei Approximationsaufgaben aus einer Menge von möglichen Problemlösungen die möglichst beste Lösung zu finden. Nun stellt sich noch die Frage: Was versteht man unter der „besten“ Lösung?

Um die Qualität einer Lösung zu beurteilen, wird ein Distanzmaß definiert, auf welches in Kapitel 2 noch näher eingegangen wird. Kapitel 2 behandelt im Weiteren die für den Verlauf der Arbeit wichtigen Definitionen und Sätze.

In Kapitel 3 werden zwei verschiedene Arten von Distanzen beschrieben, die funktionelle und die geometrische Distanz. Es werden mehrere Anwendungsgebiete der zwei Distanzen angeführt und auch auf die einzelnen Vor- und Nachteile eingegangen. Die weitere Arbeit beschäftigt sich hauptsächlich mit der geometrischen Distanz.

Mittels eines Beispiels wird in Kapitel 4 auf die einfachste Form der Approximation eingegangen: Das Annähern von Punktmengen im euklidischen Raum mittels einer Geraden. Im Laufe des Beispiels wird ein möglicher Lösungsweg verwendet, der sich dann schließlich als Hauptkomponentenanalyse herausstellt.

Der mathematische Hintergrund der Hauptkomponentenanalyse, sowie die Korrektheit der Lösung beim Approximieren mit einer Geraden oder Ebene, wird im Weiteren dargelegt, um schließlich die Korrektheit des gegebenen Beispiels einer Approximation mit Hilfe einer Geraden zu beweisen.

1 Einleitung

Ein weiterer interessanter Zusammenhang zwischen der Hauptkomponentenanalyse mit Verwendung der Eigenwertzerlegung und mit Verwendung der Singulärwertzerlegung wird in Kapitel 4 aufgezeigt.

Um sinnvoll mit Ausreißern in Datenmengen umgehen zu können, wird noch auf robuste Hauptkomponentenanalyse eingegangen. Hier werden zwei verschiedene Herangehensweisen beschrieben, einerseits die Hauptkomponentenanalyse mit Gewichtiterationen und andererseits das Zentrieren der Daten mittels $\| \cdot \|_1$ -Norm.

Der nächste große Teil der Arbeit beschäftigt sich mit der nichtlinearen Form der Methode der kleinsten Quadrate. Hier werden in Kapitel 5 drei iterative Lösungsmethoden solcher Probleme vorgestellt: die Newton-Methode, das Gauss-Newton-Verfahren und das Levenberg-Marquardt-Verfahren.

Um die Anwendung dieser drei Verfahren zu verstehen, gehen wir auch noch kurz auf Lösungsmethoden für lineare Gleichungssysteme ein, da diese für die drei Verfahren benötigt werden. Hier wird die Anwendung der Cholesky-Zerlegung, der QR-Zerlegung und der Singulärwertzerlegung beschrieben.

Anschließend wird die Methode der kleinsten Quadrate auf die Approximation eines Kreises in eine Punktmenge angewendet, um die Grundzüge verstehen zu lernen.

Kapitel 6 beschäftigt sich mit der Anwendung der Methode der kleinsten Quadrate auf ein praktisches Beispiel aus dem Bereich der Messtechnik. Hier wird das Approximieren von Profilen von Schneidkanten mittels geometrischen Objekten beschrieben, wobei besonders auf die Zuordnung der Datenpunkte zu den einzelnen Abschnitten des Schneidkantenprofils eingegangen wird. Zum Abschluss des Kapitels werden noch die Ergebnisse der vorgestellten Approximationsmethode dargelegt.

2 Definitionen

Um näher auf die verschiedenen Approximationsmethoden eingehen zu können, benötigen wir zuerst noch einige Definitionen. Diese orientieren sich hauptsächlich an den Lehrbüchern von Klein [16] und Ahn [2].

Als Grundlage für das Approximieren von Punktmenge mittels geometrischer Objekte dient die euklidische Geometrie. Um Objekte miteinander vergleichen zu können betrachten wir eine Distanzfunktion.

Definition 2.1. Sei M eine nichtleere Menge und $d : M \times M \rightarrow \mathbb{R}$ eine Funktion mit den Eigenschaften:

$$d(p, q) \geq 0 \text{ und } d(p, q) = 0 \text{ für } p = q \text{ (Definitheit)}$$

$$d(p, q) \leq d(p, r) + d(r, q) \text{ (Dreiecksungleichung)}$$

für alle p, q, r aus M , dann heißt d Halbmetrik.

Definition 2.2. Ist M eine nichtleere Menge und $d : M \times M \rightarrow \mathbb{R}$ eine Halbmetrik mit der zusätzlichen Eigenschaft

$$d(p, q) = d(q, p) \text{ (Symmetrie)}$$

für alle $p, q \in M$, dann heißt d Metrik und (M, d) heißt metrischer Raum.

Definition 2.3. Seien $p, q \in \mathbb{R}^n$ mit $p = (p_1, \dots, p_n)$ und $q = (q_1, \dots, q_n)$. Die Minkowski-Metrik $\|\cdot\|_d$ wird definiert als:

$$\|\cdot\|_d := \begin{cases} (\sum_{k=1}^n (p_k - q_k)^d)^{\frac{1}{d}} & \text{für } 1 \leq d < \infty \\ \max_{k=1}^n |p_k - q_k| & \text{für } d = \infty \end{cases}$$

Zu den wohl bekanntesten Metriken gehören drei Spezialfälle der Minkowski-Metriken:

$$\|\cdot\|_1 \text{ (Manhattan-Metrik)}, \|\cdot\|_2 \text{ (euklidische Metrik)}, \|\cdot\|_\infty \text{ (Chebyshev-Metrik)}$$

Den euklidischen Raum definiert man über die zweite dieser Metriken, die auch euklidische Metrik genannt wird:

$$\|p - q\|_2 = \sqrt{\sum_{k=1}^n (p_k - q_k)^2}$$

2 Definitionen

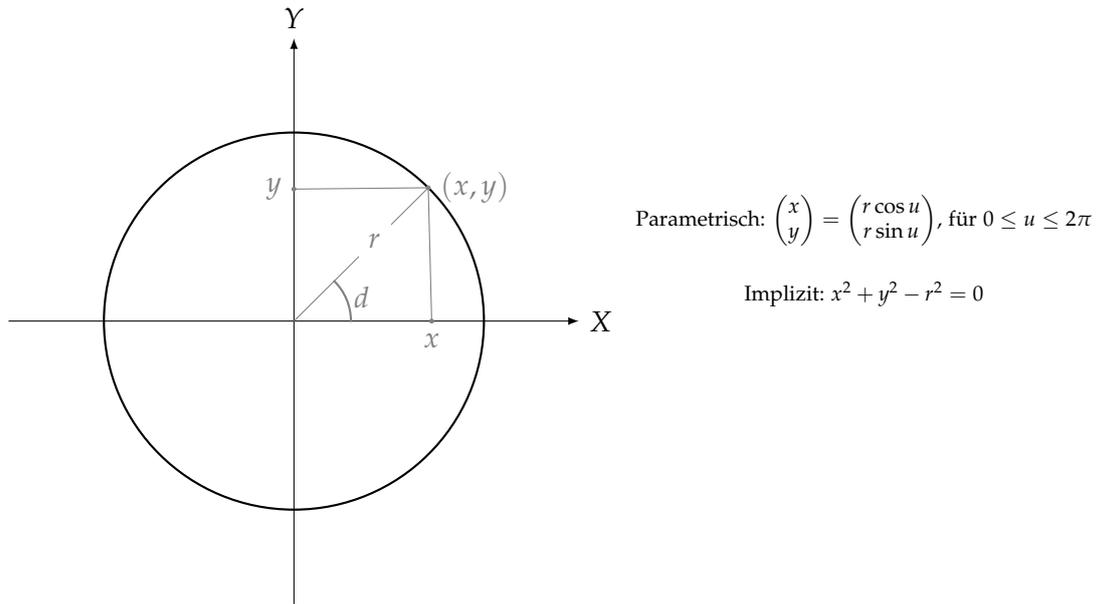


Abbildung 2.1: Darstellungen eines Kreises

für $p, q \in \mathbb{R}^n$ mit $p = (p_1, \dots, p_n)$, $q = (q_1, \dots, q_n)$. $(\mathbb{R}^n, \|\cdot\|_2)$ bezeichnet man als euklidischen Raum.

Diese Arbeit beschäftigt sich hauptsächlich mit dem \mathbb{R}^2 und \mathbb{R}^3 .

Für die Darstellung von Kurven und Flächen im \mathbb{R}^2 und \mathbb{R}^3 gibt es zwei verschiedene Möglichkeiten:

- Die implizite Darstellung
- und die Parameterdarstellung.

Als implizite Darstellung einer Kurve oder Fläche im \mathbb{R}^2 und \mathbb{R}^3 versteht man eine Gleichung der Form

$$G(\vec{a}, x, y) = 0 \text{ bzw. } G(\vec{a}, x, y, z) = 0,$$

wobei \vec{a} die Modellparameter darstellen.

In der Parameterdarstellung werden die Variablen mittels Funktionsgleichungen beschrieben, die gemeinsame Parameter durchlaufen. Eine Kurve im \mathbb{R}^2 besitzt somit die Parameterdarstellung $\begin{pmatrix} x(\vec{a}, t) \\ y(\vec{a}, t) \end{pmatrix}$ für den Parameter t und eine Fläche im \mathbb{R}^3 die

Darstellung $\begin{pmatrix} x(\vec{a}, u, v) \\ y(\vec{a}, u, v) \\ z(\vec{a}, u, v) \end{pmatrix}$ für Parameter u und v .

Eine genauere Diskussion der einzelnen Vor- und Nachteile der zwei Darstellungen kann bei vielen Autoren nachgelesen werden (siehe zum Beispiel [6], [10] oder [27]).

2.1 Lineare Algebra

Voraussetzung für die folgenden Definitionen ist ein grundlegendes Verständnis der Matrizenrechnung. Die folgenden Definitionen können in vielen Lehrbüchern im Bereich der Linearen Algebra nachgelesen werden. Wir orientieren uns jedoch hauptsächlich an Jauregui [14] und Gentle [12].

In diesem Abschnitt betrachten wir eine Zerlegungsmethode einer Matrix, die für die folgenden Kapitel eine große Rolle spielt. Diese Methode wird als Singulärwertzerlegung bezeichnet.

Um die Existenz der Singulärwertzerlegung einer Matrix zu zeigen, benötigt man etwas Vorarbeit:

Satz 2.4. *Sei A eine symmetrische $(n \times n)$ -Matrix, dann ist A orthogonal diagonalisierbar und besitzt nur reelle Eigenwerte, d.h. es existieren reelle Zahlen $\lambda_1, \dots, \lambda_n$ und orthogonale, reelle Vektoren $\vec{v}_1, \dots, \vec{v}_n$, so dass $\forall i = 1, \dots, n$:*

$$A\vec{v}_i = \lambda_i\vec{v}_i.$$

Um diesen Satz anwenden zu können, muss also die Voraussetzung, dass die Matrix A symmetrisch ist, gegeben sein. Hat man nun eine $(n \times m)$ -Matrix A mit $n \neq m$ gegeben, kann man diesen Satz jedoch noch immer auf die Matrizen $A^T A$ und AA^T anwenden, da diese symmetrisch sind.

Nun stellt sich die Frage, inwiefern die Eigenwerte und Eigenvektoren der zwei Matrizen $A^T A$ und AA^T miteinander verwandt sind.

Lemma 2.5. *Die Matrizen $A^T A$ und AA^T besitzen die selben, von Null verschiedenen Eigenwerte.*

Beweis. Sei \vec{v} ein vom Nullvektor verschiedener Eigenvektor von $A^T A$ mit Eigenwert $\lambda \neq 0$, also

$$(A^T A)\vec{v} = \lambda\vec{v}.$$

Multiplikation mit A von links ergibt

$$AA^T(A\vec{v}) = \lambda(A\vec{v}).$$

Damit λ ein gültiger Eigenwert ist, muss nun nur noch überprüft werden, ob $A\vec{v}$ nicht der Nullvektor ist, da Eigenvektoren von Null verschieden sein müssen. Dies gilt jedoch, da ansonsten $\lambda\vec{v} = 0$ in der ersten Gleichung gelten müsste, was jedoch ein Widerspruch zu $\vec{v} \neq \vec{0}$ und $\lambda \neq 0$ ist. Dies bedeutet also, dass $A\vec{v}$ ein Eigenvektor von AA^T mit Eigenwert λ ist.

Die Umkehrung gilt ebenso. □

2 Definitionen

Beispiel 2.6. Um dieses Ergebnis nun an einem Beispiel anzuwenden, betrachtet man eine (1000×3) -Matrix A . Man kann nun also das Eigenwertproblem der (1000×1000) -Matrix AA^T auf das Berechnen der Eigenwerte der (3×3) -Matrix $A^T A$ reduzieren, da die restlichen 997 Eigenwerte der Matrix AA^T gleich Null sind.

Es ist also möglich, jede symmetrische Matrix mittels ihrer Eigenwerte und Eigenvektoren zu beschreiben. Um diese Darstellung nun auf nicht symmetrische Matrizen zu erweitern, betrachtet man die Singulärwertzerlegung einer Matrix. Die folgenden Definitionen orientieren sich an Richter und Wick [24].

Definition 2.7. Gegeben sei eine Matrix $A \in \mathbb{R}^{m \times n}$. Man bezeichnet die Darstellung

$$A = U\Sigma V^T$$

als die Singulärwertzerlegung von A , wenn $U \in \mathbb{R}^{m \times m}$ und $V \in \mathbb{R}^{n \times n}$ orthogonale Matrizen sind, sowie für die Einträge der Matrix $\Sigma = (s_{i,j}) \in \mathbb{R}^{m \times n}$ gilt, dass $s_{i,j} = 0 \forall i \neq j$ und die Diagonaleinträge $s_{1,1} \geq s_{2,2} \geq \dots$ nicht negativ sind. Die Einträge in U werden als linke Singulärvektoren und die Einträge in V als rechte Singulärvektoren bezeichnet. Die Diagonaleinträge der Matrix Σ

$$\sigma_i := s_{i,i}$$

heißen Singulärwerte von A .

Beispiel 2.8.

$$A = \begin{pmatrix} 0.36 & 1.16 & 0.48 \\ 0.48 & -1.20 & 0.64 \end{pmatrix} = \begin{pmatrix} 0.8 & 0.6 \\ -0.6 & 0.8 \end{pmatrix} \begin{pmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 \\ 0.6 & 0 & 0.8 \\ -0.8 & 0 & 0.6 \end{pmatrix}.$$

Bemerkung 2.9. Die Anzahl der von Null verschiedenen Singulärwerte von A ist gleich dem Rang von A .

Satz 2.10. Existiert eine Singulärwertzerlegung einer Matrix $A \in \mathbb{R}^{n \times m}$ in orthogonale Matrizen $U \in \mathbb{R}^{n \times n}$, $V \in \mathbb{R}^{m \times m}$ und eine erweiterte Diagonalmatrix $\Sigma \in \mathbb{R}^{n \times m}$, dann entsprechen die Singulärwerte den Wurzeln der Eigenwerte der Matrix $A^T A$ und sind bis auf Permutation eindeutig bestimmt.

Beweis. Existiert eine Singulärwertzerlegung, gilt mit den orthogonalen Matrizen U und V

$$AV = U\Sigma \text{ und } A^T U = V\Sigma.$$

Seien nun $V = (\vec{v}_1, \dots, \vec{v}_m)$ und $U = (\vec{u}_1, \dots, \vec{u}_n)$ mit Spaltenvektoren \vec{v}_i und \vec{u}_i und σ_i die Diagonaleinträge der Matrix Σ , dann gilt

$$A\vec{v}_i = \sigma_i \vec{u}_i \quad \forall i \in \{1, \dots, m\},$$

$$A^T \vec{u}_i = \sigma_i \vec{v}_i \quad \forall i \in \{1, \dots, m\}.$$

2 Definitionen

Hieraus folgt mit $p = \min\{n, m\}$, dass

$$A^T A \vec{v}_i = \sigma_i A^T \vec{u}_i = \sigma_i^2 \vec{v}_i \quad \forall i \in \{1, \dots, p\}.$$

Somit sind die σ_i^2 Eigenwerte der Matrix $A^T A$. Dass die Matrix $A^T A$ keine weiteren Eigenwerte haben kann, folgt für $n \geq m$, da die Anzahl der σ_i der Größe der Matrix $A^T A$ entspricht, und für $n < m$ aus Lemma 2.5. Die Eindeutigkeit folgt aus der Eindeutigkeit der Eigenwerte der Matrix $A^T A$. \square

Dies zeigt, dass man die Existenz der Singulärwertzerlegung einer Matrix A mittels Konstruktion über die Eigenwerte der Matrix $A^T A$ zeigen kann.

Satz 2.11. Gegeben sei eine Matrix $A \in \mathbb{R}^{m \times n}$. Dann existiert eine Singulärwertzerlegung

$$A = U \Sigma V^T$$

von A , wobei $V = (\vec{v}_1, \dots, \vec{v}_m)$ und $U = (\vec{u}_1, \dots, \vec{u}_n)$ mit Spaltenvektoren \vec{v}_i und \vec{u}_i und λ die Diagonaleinträge der Matrix Σ sind.

Mit Hilfe der Singulärwertzerlegung lässt sich A auch darstellen als

$$\sum \sqrt{\lambda_i} \vec{u}_i \vec{v}_i^T.$$

Beweis. Man betrachte die symmetrische $(n \times n)$ -Matrix $A^T A$ und bestimme ihre Eigenwerte λ_i und eine Orthonormalbasis von Eigenvektoren \vec{v}_i , wobei $\lambda_1 \leq \dots \leq \lambda_n$. Da einerseits

$$\vec{v}_i^T A^T A \vec{v}_i = \lambda_i \vec{v}_i^T \vec{v}_i = \lambda_i,$$

und andererseits

$$\vec{v}_i^T A^T A \vec{v}_i = (A \vec{v}_i)(A \vec{v}_i) \geq 0$$

gilt, sind alle λ_i nichtnegativ. Da der Rang von $A^T A$ gleich dem Rang von A ist, sind genau die ersten r Eigenwerte $\lambda_1, \dots, \lambda_r$ positiv. Setze

$$\vec{u}_i = \frac{1}{\sqrt{\lambda_i}} A \vec{v}_i, \quad \text{für } i = 1, \dots, r$$

und bestimme $m - r$ orthonormale Vektoren $\vec{u}_{r+1}, \dots, \vec{u}_m$, die zu $\vec{u}_1, \dots, \vec{u}_r$ orthogonal sind. Aus diesen Vektoren lassen sich nun die Matrizen

$$U = (\vec{u}_1, \dots, \vec{u}_m) \quad \text{und}$$

$$V = (\vec{v}_1, \dots, \vec{v}_n)$$

bilden. Mittels der λ_i stellt man die Matrix $\Sigma = (s_{i,j}) \in \mathbb{R}^{m \times n}$ auf, wobei

$$s_{i,j} := \begin{cases} \sqrt{\lambda_i} & \text{für } i = j \leq r \\ 0 & \text{sonst.} \end{cases}$$

Die Gleichung $A = U \Sigma V^T$ ist auf Grund der Konstruktion erfüllt und kann leicht nachgerechnet werden. Es fehlt noch, zu zeigen, dass $A = U \Sigma V^T$ auch wirklich eine Singulärwertzerlegung von A ist:

2 Definitionen

- $\{\vec{v}_1, \dots, \vec{v}_n\}$ ist nach Konstruktion ein Orthonormalbasis von \mathbb{R}^n , also ist V eine orthogonale Matrix.
- $\{\vec{u}_1, \dots, \vec{u}_m\}$ ist eine Orthonormalbasis von \mathbb{R}^m , da für $i, j = 1, \dots, r$ gilt

$$\vec{u}_i^T \vec{u}_j = \frac{1}{\sqrt{\lambda_i \lambda_j}} \vec{v}_i^T A^T A \vec{v}_j = \frac{\sqrt{\lambda_j}}{\sqrt{\lambda_i}} \vec{v}_i^T \vec{v}_j = \begin{cases} \sqrt{\frac{\lambda_j}{\lambda_i}} = 1 & \text{für } i = j \\ 0 & \text{sonst} \end{cases}.$$

Also sind $\vec{u}_1, \dots, \vec{u}_r$ orthonormal und auf $\vec{u}_{r+1}, \dots, \vec{u}_n$ setzt sich die Orthonormalität nach Konstruktion fort. Also ist die Matrix U orthogonal.

- Es gilt

$$\begin{aligned} \sum_{i=1}^r \sqrt{\lambda_i} \vec{u}_i \vec{v}_i^T &= \sum_{i=1}^r A \vec{v}_i \vec{v}_i^T && \text{(Definition von } \vec{u}_i) \\ &= \sum_{i=1}^n A \vec{v}_i \vec{v}_i^T && \text{(inklusive Eigenvektoren zum Eigenwert 0)} \\ &= A \sum_{i=1}^n \vec{v}_i \vec{v}_i^T \\ &= AVV^T \\ &= AI = A && (\{\vec{v}_1, \dots, \vec{v}_n\} \text{ Orthonormalbasis}) \end{aligned}$$

□

2.2 Statistik

Dieser Abschnitt deckt eine kurze Einführung in die statistischen Begriffe ab, die in den folgenden Kapiteln verwendet werden. Die Definitionen sind der Einführung von Smith [26] entnommen.

Definition 2.12. Sei $\{X_i\}_{1 \leq i \leq n} \in \mathbb{R}$ eine Menge von Daten, dann bezeichnet

$$\bar{X} = \frac{\sum_{i=1}^n X_i}{n}$$

den Mittelwert der Datenmenge. Dieser wird auch oft als Schwerpunkt bezeichnet.

Um die Streuung von Mengen zu messen, betrachtet man die durchschnittliche Distanz des Mittelwertes zu jedem Datenpunkt:

Definition 2.13. Sei $\{X_i\}_{1 \leq i \leq n} \in \mathbb{R}$ eine Menge von Daten, dann bezeichnet

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n - 1}}$$

die Standardabweichung der Datenmenge.

2 Definitionen

Ein weiterer Indikator der Streuung ist die Varianz, die sich aus der Standardabweichung ergibt:

$$\text{Var}(X) = \sigma^2 = \frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n-1}.$$

Standardabweichung und Varianz lassen sich somit für Datensätze aus \mathbb{R} berechnen. Um dies nun auf Daten aus \mathbb{R}^2 zu überführen betrachtet man die Kovarianz:

Definition 2.14. Sei $(X_i, Y_i)_{1 \leq i \leq n} \in \mathbb{R}^2$ ein Datensatz, dann bezeichnet

$$\text{Cov}(X, Y) = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{n-1}$$

die Kovarianz des Datensatzes.

Die Kovarianz beschreibt die Abhängigkeit der zwei Dimensionen. Wichtig hier ist das Vorzeichen des Ergebnisses:

Ist dieses positiv, heißt dies, dass die Variablen gemeinsam wachsen oder fallen. Ist die Kovarianz jedoch negativ, bedeutet dies, dass die Variablen in einer Dimension wachsen, während sie in der anderen fallen.

Betrachtet man nun einen n -dimensionalen Datensatz, kann man für je zwei der Dimensionen die Kovarianz berechnen. Dies führt zum Begriff der Kovarianzmatrix:

Definition 2.15. Seien Datenpunkte $(\vec{x}_1, \dots, \vec{x}_n) \in \mathbb{R}^m$ gegeben mit $\vec{x}_i = \begin{pmatrix} x_{i1} \\ \vdots \\ x_{im} \end{pmatrix}$ und dem

Mittelwert $\vec{\bar{x}} = \begin{pmatrix} \bar{x}_1 \\ \vdots \\ \bar{x}_m \end{pmatrix}$, dann ist die Kovarianzmatrix definiert als $C = (c_{i,j})$, mit

$$c_{i,j} = \frac{\sum_{d=1}^n (x_{di} - \bar{x}_i)(x_{dj} - \bar{x}_j)}{n-1}.$$

Um eine genauere Information bezüglich der Abhängigkeit zwischen den Dimensionen einer zweidimensionalen Punktmenge $(X_i, Y_i)_{1 \leq i \leq n}$ zu bekommen, kann man auch den Korrelationskoeffizienten betrachten.

Definition 2.16. Sei $(X_i, Y_i)_{1 \leq i \leq n} \in \mathbb{R}^2$ eine Punktmenge, dann nennt man

$$\text{Cor}_{X,Y} = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{(n-1)\sigma_X\sigma_Y} = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^n (X_i - \bar{X})^2 \sum_{i=1}^n (Y_i - \bar{Y})^2}}$$

den Korrelationskoeffizienten der Punktmenge.

2 Definitionen

Behauptung 2.17. Der Korrelationskoeffizient besitzt die Eigenschaft

$$\text{Cor}_{X,Y} \in [-1, 1].$$

Beweis. Betrachtet man für beliebige $u, v \in \mathbb{R}$ den Ausdruck:

$$0 \leq \frac{1}{n-1} \sum_{i=1}^n (u(X_i - \bar{X}) + v(Y_i - \bar{Y}))^2,$$

ergibt sich durch Umformungen

$$\begin{aligned} 0 &\leq \frac{1}{n-1} \sum_{i=1}^n (u(X_i - \bar{X}) + v(Y_i - \bar{Y}))^2 \\ &= \frac{1}{n-1} \sum_{i=1}^n (u^2(X_i - \bar{X})^2 + 2uv(X_i - \bar{X})(Y_i - \bar{Y}) + v^2(Y_i - \bar{Y})^2) \\ &= u^2 \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2 + 2uv \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y}) + v^2 \frac{1}{n-1} \sum_{i=1}^n (Y_i - \bar{Y})^2 \\ &= u^2 \text{Var}(X) + 2uv \text{Cov}(XY) + v^2 \text{Var}(Y), \end{aligned}$$

Setzt man nun $u = \text{Var}(Y)$ und $v = -\text{Var}(X)$, folgt

$$\text{Var}(X)\text{Var}(Y)\text{Cov}(XY) \leq \text{Var}(X)^2\text{Var}(Y)^2,$$

wodurch man

$$\text{Cov}(XY) \leq \frac{\text{Var}(X)^2\text{Var}(Y)^2}{\text{Var}(X)\text{Var}(Y)} = \text{Var}(X)\text{Var}(Y)$$

erhält. Daraus ergibt sich schließlich

$$\text{Cor}_{X,Y} = \frac{\text{Cov}(XY)}{\text{Var}(X)\text{Var}(Y)} \leq 1.$$

Für $u = \text{Var}(Y)$ und $v = \text{Var}(X)$ folgt analog

$$\text{Cor}_{X,Y} \geq -1,$$

wodurch die Aussage bewiesen ist. \square

Für den Korrelationskoeffizient gilt folgendes:

- Ein positiver Wert beschreibt eine positive Korrelation: Die Variablen wachsen/fallen gemeinsam.
- Ein negativer Wert beschreibt eine negative Korrelation: Wenn eine Variable wächst/fällt, fällt/wächst die andere.
- Ein Wert von 0 beschreibt, dass die Variablen keine Relation zueinander besitzen

2 Definitionen

Je näher der Korrelationskoeffizient bei 1 bzw. -1 ist, desto stärker ist die Korrelation zwischen den zwei Variablen.

Betrachtet man nun einen n -dimensionalen Datensatz, kann man, wie schon für die Kovarianzmatrix, für je zwei der Dimensionen den Korrelationskoeffizienten berechnen und somit die Korrelationsmatrix definieren:

Definition 2.18. Sei $(\vec{x}_1, \dots, \vec{x}_n)$ ein n -dimensionaler Datensatz, dann ist die Korrelationsmatrix definiert als $\text{Corr} = (c_{i,j})$, mit $c_{i,j}$ der Korrelationskoeffizient der Dimensionen i und j des Datensatzes.

3 Funktionelles und geometrisches Fitting

Unter „Fitting“ versteht man das Einpassen von Objekten oder Funktionen in Punktwolken. Dieser Bereich teilt sich in Interpolation, das exakte Einpassen in eine Menge von Punkten, und Approximation auf (siehe Abbildung 3.1). Im weiteren Text wird unter „Fitting“ das approximative Einpassen verstanden.

In der Literatur wird zwischen zwei unterschiedlichen Approximations-Methoden differenziert: die funktionelle und die geometrische Approximation. Diese unterscheiden sich in der Definition ihrer Fehlerfunktionen, die minimiert werden sollen.

Unter funktioneller Approximation im \mathbb{R}^2 versteht man normalerweise den Versuch, das Zielobjekt so in eine Punktmenge einzupassen, dass der vertikale (oder auch der horizontale) Abstand zwischen den einzelnen Punkten und dem Objekt minimal wird. In der Statistik kennt man das funktionelle Fitten von Geraden und Kreisen auch unter dem Begriff lineare Regression. In diesem Fall sind also k Punkte im \mathbb{R}^2

$$\{(x_1, y_1), \dots, (x_k, y_k)\}$$

gegeben, und es wird versucht, eine lineare Funktion $y = f(x)$ so in die Menge an Punkten einzupassen, dass die Quadratsumme der vertikalen Abstände zwischen den Punkten und der Funktion minimal wird. Tritt jedoch der Fall auf, dass die gegebenen Punkte vertikal ausgerichtet sind, bekommt man mit dieser Methode kein Ergebnis.

Allgemein definiert man die funktionelle Distanz wie folgt:

Gegeben sei die Gleichung einer Kurve

$$F(\vec{a}, \vec{X}) \doteq \sum_{j=1}^p a_j x^{k_j} y^{l_j} = 0 \text{ mit } 0 \leq k_j + l_j \leq g \quad \forall j$$

oder einer Fläche

$$F(\vec{a}, \vec{X}) \doteq \sum_{j=1}^p a_j x^{k_j} y^{l_j} z^{m_j} = 0 \text{ mit } 0 \leq k_j + l_j + m_j \leq g \quad \forall j,$$

die von Parametern \vec{a} abhängt, wobei g den Grad der Funktion darstellt. Sei im Weiteren $\{X_i\}_{1 \leq i \leq m}$ eine Menge an Punkten. Liegt ein Punkt X_i nicht am gegebenen Modell, erhält man mittels Einsetzen des Punktes in die Funktion

$$F(\vec{a}, \vec{X}_i) \neq 0.$$

3 Funktionelles und geometrisches Fitting

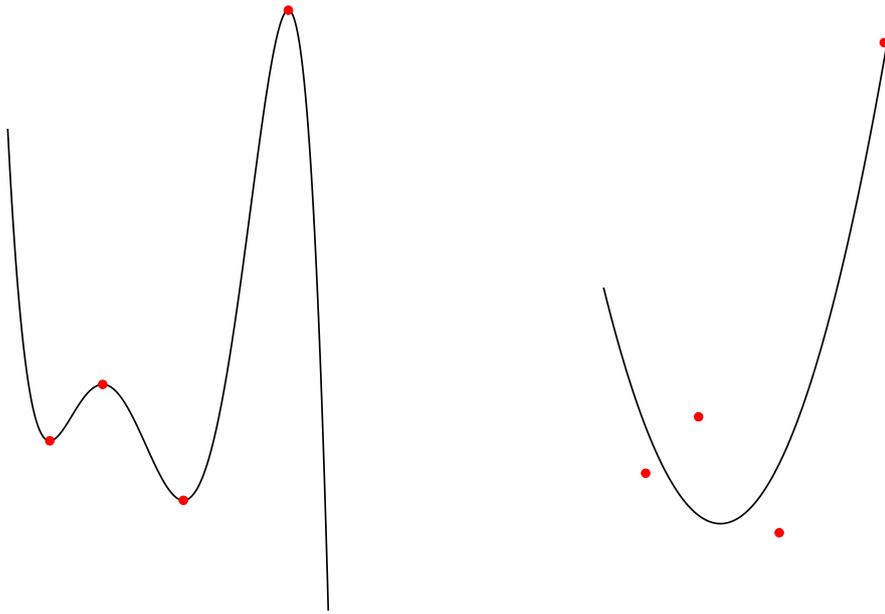


Abbildung 3.1: Unterschied zwischen Interpolation (links) und Approximation (rechts)

Dieses Fehlermaß $F(\vec{a}, \vec{X}_i)$ wird als funktionelle Distanz zwischen dem Punkt X_i und der Kurve $F(\vec{a}, \vec{X})$ bezeichnet.

Man versucht also Parameter \vec{a} zu finden, die die Quadratsumme der funktionellen Distanzen

$$\sum_{k=1}^m F(\vec{a}, X_k)^2$$

zu gegebenen Punkten $\{X_i\}_{1 \leq i \leq m}$ minimiert.

Um die triviale Lösung $\vec{a} = \vec{0}$ zu vermeiden, muss \vec{a} eine Nebenbedingung erfüllen. Hierfür wird meist die Gleichung

$$\|\vec{a}\| = 1$$

verwendet [11].

Um das Ergebnis der funktionellen Approximation zu verbessern, wird auch oft die normalisierte funktionelle Distanz

$$\sum_{k=1}^m \left(\frac{F(\vec{a}, X_k)}{\|\nabla F(\vec{a}, X_k)\|} \right)^2$$

verwendet, wobei $\nabla = (\partial/\partial x, \partial/\partial y, \partial/\partial z)^T$.

3 Funktionelles und geometrisches Fitting

Bei der geometrischen Approximation hingegen, auch bekannt als „Orthogonal Distance Fitting“, betrachtet man die kürzeste Distanz zwischen den Punkten und dem Objekt. Man versucht also den Abstand des Objektes zu den gegebenen Punkten zu minimieren

$$\min \sum_{k=1}^m \|X_k - X'_k\|^2,$$

wobei $\{X_k\}_{1 \leq k \leq m}$ die gegebenen Punkte sind, die man mittels des gegebenen Objektes approximieren möchte, $\{X'_k\}_{1 \leq k \leq m}$ die zugehörigen nächsten Punkte am Objekt und die verwendete Norm die euklidische Norm ist.

Bevor wir näher auf die Verwendung der geometrischen Approximation eingehen, muss zuerst die Existenz der Punkte $\{X'_k\}_{1 \leq k \leq m}$ bewiesen werden. Hierfür betrachten wir folgende Behauptung:

Behauptung 3.1. *Sei $O \neq \emptyset$ eine abgeschlossene Menge und X ein Punkt im \mathbb{R}^n , dann existiert ein Punkt $X_0 \in O$, so dass*

$$\|X - X_0\| \leq \|X - \bar{X}\| \quad \forall \bar{X} \in O.$$

Beweis. Da $O \neq \emptyset$, folgt, dass ein Punkt $X^* \in O$ existiert. Damit ist die minimale Distanz von O zum gegebenen Punkt X kleiner oder gleich $\|X^* - X\|$.

Definiere nun die Menge $O' = O \cap \{Y : \|X - Y\| \leq \|X - X^*\|\}$. Da O' abgeschlossen und beschränkt ist und $\|\cdot\|$ eine stetige Funktion darstellt, folgt mit dem Satz von Weierstrass, dass ein Punkt $X_0 \in O'$ existiert, so dass

$$\|X - X_0\| = \text{Inf}\{\|X - Y\|, Y \in O'\}.$$

Auf Grund der Definition von O' folgt, dass $X_0 \in O$, womit die Existenz bewiesen wäre. □

Die geometrische Distanz wird auch sehr oft in der Koordinaten-Messtechnik verwendet [1].

Die Verwendung der geometrischen Distanz benötigt bei vielen Objekten nichtlineare Rechnungen, die das Lösen solcher Aufgaben sowohl analytisch als auch rechnerisch erschweren. Eine Ausnahme hierfür bildet das Approximieren mittels einer Geraden oder einer Fläche, für die solche Probleme explizit lösbar sind (siehe Abschnitt 4.1). Stellt man nun also die zwei Verfahren gegenüber, steht man vor einer schwierigen Entscheidung: Nimmt man lieber den hohen Rechenaufwand in Kauf, um einige Nachteile der funktionellen Distanz nicht hinnehmen zu müssen, oder versucht man, seine Rechenzeit minimal zu halten?

Vergleicht man nun das Approximieren einer Menge von Punkten im \mathbb{R}^2 mittels einer Geraden mit Hilfe der geometrischen Distanz, der funktionellen Distanz in vertikaler

3 Funktionelles und geometrisches Fitting

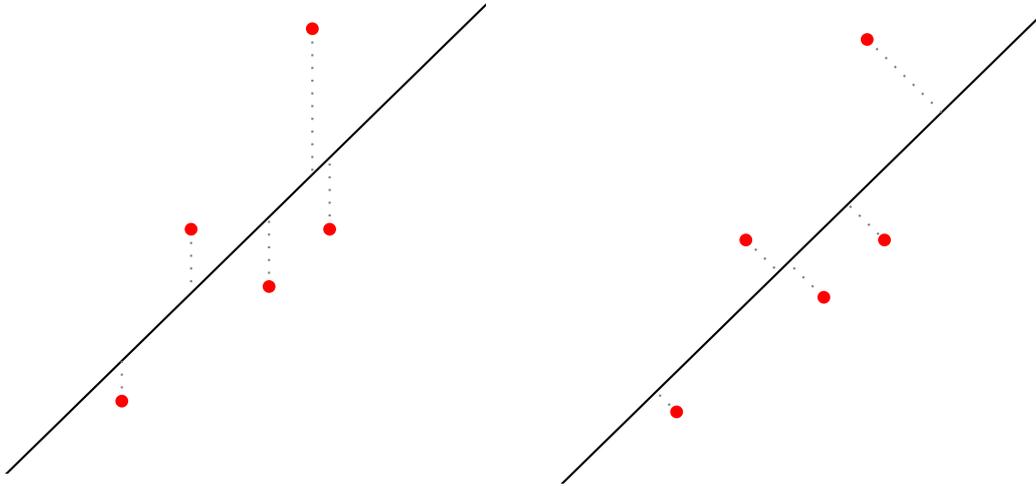


Abbildung 3.2: Unterschied zwischen funktioneller (links) und geometrischer (rechts) Approximation

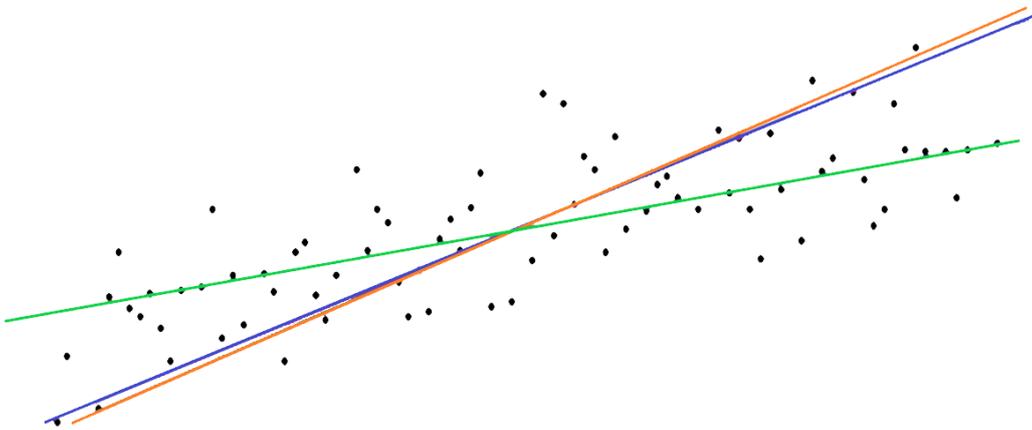


Abbildung 3.3: Approximation einer Punktmenge mit einer Geraden mittels funktioneller Distanz in X (grün), in Y (orange) und mittels geometrischer Distanz (blau) im \mathbb{R}^2

3 Funktionelles und geometrisches Fitting

Richtung und der funktionellen Distanz in horizontaler Richtung, sind hier einige Unterschiede ersichtlich (siehe Abbildung 3.3).

Die Steigungen der Geraden, welche mit der funktionellen Distanz in horizontaler Richtung, und jener, deren Abstände zu den Punkten in vertikaler Richtung minimiert wurde, sind sehr unterschiedlich voneinander. Im Weiteren sieht man, dass die Gerade, die mit Hilfe der geometrischen Distanz eingepasst wurde, für diese Punktmenge ähnlich zur Geraden ist, welche mit der funktionellen Distanz in horizontaler Richtung eingepasst wurde.

Auf Grund des Unterschiedes in der Komplexität der Lösung dieser zwei Verfahren, werden hier ein paar Methoden vorgestellt, wie man Approximationsprobleme mit Hilfe von geometrischen Distanzen lösen kann.

Zu Beginn des nächsten Kapitels betrachten wir das einfachste geometrische Approximationsproblem, nämlich das Approximieren einer Punktmenge im \mathbb{R}^2 mittels einer Geraden, und geben einen Lösungsweg an.

4 Approximation mit einfachen Objekten

Im folgenden Kapitel beschäftigen wir uns mit dem Approximieren von Punktwolken mittels einfacher geometrischer Objekte. Diese Art der Approximation spielt eine große Rolle im Bereich der Messtechnik und CAD/CAM. Hier setzt man sich zum Ziel, unter anderem die Form und Größe von Messobjekten mit Hilfe von approximierenden geometrischen Objekten zu beurteilen.

Den verwendeten Algorithmen liegt meist die Methode der kleinsten Quadrate zugrunde, welcher die Modellparameter mittels der Minimierung der Quadratsumme einer vordefinierten Fehlerfunktion berechnet. [2]

4.1 Geometrische Approximation mit Hilfe einer Geraden im \mathbb{R}^3

Das Problem der geometrischen Approximation mittels einer Geraden oder Fläche wurde schon 1901 von K. Pearson [23] gelöst. Obwohl diese zu den einfachsten Problemen in der Klasse der geometrischen Approximationen gehören, sollte man sie nicht übergehen, da diese zum Verständnis des Lösens solcher Probleme beitragen.

Diese Arten der Approximation finden besonders Anwendung in der geometrischen Datenverarbeitung und Statistik.

Betrachtet man nun m Punkte $\{X_i\}_{1 \leq i \leq m} \in \mathbb{R}^3$ und eine Gerade in Parameterform

$$X_s + u \cdot \vec{r} \text{ mit } \|\vec{r}\| = 1 \text{ und } u \in \mathbb{R},$$

mit einem auf ihr liegenden Punkt X_s und einem Einheitsvektor \vec{r} , bekommt man für den orthogonalen Abstand eines beliebigen Punktes X_i zu der gegebenen Geraden

$$\|(X_i - X_s) \times \vec{r}\|$$

und bildet damit die Fehlerfunktion

$$F(\vec{a}, X_i) = \sum_{i=1}^m \|(X_i - X_s) \times \vec{r}\|^2,$$

die es zu minimieren gilt, wobei $\vec{a} = (X_s, \vec{r})$.

4 Approximation mit einfachen Objekten

Betrachtet man eine Punktmenge im Raum, ist es nicht überraschend, dass eine geometrisch eingepasste Gerade durch den Schwerpunkt der Punkte geht (siehe Lemma 4.2). Dieser ist gegeben durch

$$X_s = \bar{X} = \frac{1}{m} \left(\sum_{i=1}^m x_i, \sum_{i=1}^m y_i, \sum_{i=1}^m z_i \right)$$

mit $X_i = (x_i, y_i, z_i)$.

Um nun noch die Richtung der Geraden zu bestimmen, und den kleinstmöglichen Fehler zu erhalten, betrachtet man die Kovarianzmatrix der Punktmenge $\{X_i\}_{1 \leq i \leq m}$. Mit $\bar{X} = (\bar{x}, \bar{y}, \bar{z})$ ersetzen wir, der Einfachheit halber, zuerst

$$x_i = x_i - \bar{x}, y_i = y_i - \bar{y} \text{ und } z_i = z_i - \bar{z}.$$

Dies führt zu der zugehörigen Kovarianzmatrix:

$$H = \begin{pmatrix} \text{Cov}(X, X) & \text{Cov}(X, Y) & \text{Cov}(X, Z) \\ \text{Cov}(Y, X) & \text{Cov}(Y, Y) & \text{Cov}(Y, Z) \\ \text{Cov}(Z, X) & \text{Cov}(Z, Y) & \text{Cov}(Z, Z) \end{pmatrix} =$$

$$\frac{1}{m-1} \begin{pmatrix} \sum_{i=1}^m x_i^2 & \sum_{i=1}^m x_i \cdot y_i & \sum_{i=1}^m x_i \cdot z_i \\ \sum_{i=1}^m y_i \cdot x_i & \sum_{i=1}^m y_i^2 & \sum_{i=1}^m y_i \cdot z_i \\ \sum_{i=1}^m z_i \cdot x_i & \sum_{i=1}^m z_i \cdot y_i & \sum_{i=1}^m z_i^2 \end{pmatrix}.$$

Diagonalisiert man diese Matrix, bekommt man als Ergebnis folgende Faktorisierung der Kovarianzmatrix

$$H = V_H W_H V_H^T,$$

wobei die Matrix W die Eigenwerte auf ihrer Diagonale hat und die Matrix V aus orthogonalen Vektoren, den sogenannten Hauptkomponenten, besteht, also

$$W_H = \text{diag}(w_1, w_2, w_3),$$

$$V_H = (\vec{v}_1, \vec{v}_2, \vec{v}_3), \text{ mit } V_H^T V_H = I.$$

Wählen wir nun \vec{v}_j mit $j \in \{1, 2, 3\}$ und w_j maximal, bekommen wir die Lösung für das geometrische Approximationsproblem, nämlich die Gerade, die den Mittelwert der Punktmenge enthält und in Richtung der ersten Hauptkomponente zeigt (siehe Satz 4.4):

$$\bar{X} + u \cdot \vec{v}_j.$$

Betrachten wir dies nun an einem Beispiel und wählen, der Anschaulichkeit halber, Punkte aus dem \mathbb{R}^2 . In Tabelle 4.1 wurden zehn zufällige Punkt im Raum gewählt, in die eine Gerade gefittet werden soll.

Daraus ergibt sich der Mittelwert $\bar{X} = (1.47, 1.22)$, der nun von allen Punkten abgezogen werden muss, um den Schwerpunkt der Punktmenge in den Ursprung zu verschieben. Somit ergibt sich Tabelle 4.2 mit den neuen Punkten.

4 Approximation mit einfachen Objekten

Tabelle 4.1: Zufällig verteilte Punkte im \mathbb{R}^2

| | p1 | p2 | p3 | p4 | p5 | p6 | p7 | p8 | p9 | p10 |
|---|-----|-----|----|------|------|------|----|-----|-----|-----|
| x | 2.3 | 1.9 | 0 | 2.4 | 0.6 | 0 | 1 | 2.3 | 1.5 | 2.7 |
| y | -2 | 8.1 | -3 | -4.1 | -1.1 | -0.3 | 1 | 3.3 | 5 | 5.3 |

Tabelle 4.2: In den Ursprung verschobene Punktmenge

| | p1 | p2 | p3 | p4 | p5 | p6 | p7 | p8 | p9 | p10 |
|---|-------|------|------|-------|-------|-------|-------|-------|------|------|
| x | 0.83 | 1.23 | 0.43 | -1.47 | 0.93 | -0.87 | -1.47 | -0.47 | 0.83 | 0.03 |
| y | -3.22 | 4.08 | 6.88 | -4.22 | -5.32 | -2.32 | -1.52 | -0.22 | 2.08 | 3.78 |

Mit Hilfe der verschobenen Punkte erhalten wir die Kovarianzmatrix

$$H = \begin{pmatrix} 9.241 & 12.756 \\ 12.756 & 146.816 \end{pmatrix},$$

und diagonalisieren sie:

$$\begin{aligned} H &= V_H W_H V_H^T \\ &= \begin{pmatrix} 0.0915505 & 0.9958 \\ 0.9958 & -0.0915505 \end{pmatrix} \begin{pmatrix} 147.989 & 0 \\ 0 & 8.06826 \end{pmatrix} \begin{pmatrix} 0.0915505 & 0.9958 \\ 0.9958 & -0.0915505 \end{pmatrix}. \end{aligned}$$

Der größte Eigenwert ist somit 147.989, wodurch die Richtung der gefitteten Geraden

$$\vec{v}_1 = \begin{pmatrix} 0.0915505 \\ 0.9958 \end{pmatrix}$$

in der ersten Zeile der Matrix V_H abzulesen ist. Somit ergibt sich die Gerade

$$\bar{X} + u * v_1 = \begin{pmatrix} 1.47 \\ 1.22 \end{pmatrix} + u \begin{pmatrix} 0.0915505 \\ 0.9958 \end{pmatrix}.$$

In Abbildung 4.1 sieht man das Ergebnis der approximierten Geraden in der gegebenen Punktmenge.

Das oben verwendete Verfahren kann im \mathbb{R}^n mit $n \in \mathbb{N}$ beliebig verwendet werden und ist auch als „Hauptkomponentenanalyse“ oder im Englischen als „principal component analysis“, kurz PCA, bekannt. Dem Beweisen der Korrektheit diese Verfahrens widmen sich die nächsten Abschnitte.

4.2 Hauptkomponentenanalyse

Die Hauptkomponentenanalyse, im Folgenden PCA genannt, ist eine Methode, Muster in einer Menge von Daten zu erkennen. Dies ist besonders hilfreich, wenn man

4 Approximation mit einfachen Objekten

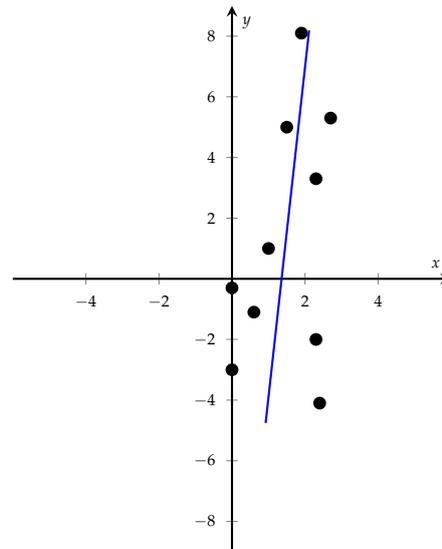


Abbildung 4.1: Ergebnis der Approximation mittels einer Geraden im \mathbb{R}^2

mehrdimensionale Daten hat, die man visuell nicht mehr analysieren kann. In diesem Fall kann man mittels PCA die Anzahl der Dimensionen reduzieren, indem man jene Dimensionen filtert, die die geringste Bedeutung für den aktuellen Datensatz besitzen.

PCA versucht eine Menge von $k < d$ Einheitsvektoren $\vec{v}_i \in \mathbb{R}^d$ für $i = 1, \dots, k$ zu finden, so dass die Varianz der Daten projiziert auf die Richtung \vec{v}_i maximiert wird, und weiters $\forall i$ gilt, dass \vec{v}_i orthogonal zu $\vec{v}_1, \dots, \vec{v}_{i-1}$ ist.

Um diese Einheitsvektoren für eine Menge von d -dimensionalen Daten $\{X_i\}_{1 \leq i \leq n}$ mit $X_i = (x_{i1}, \dots, x_{id}) \forall i = 1, \dots, n$ zu finden, verschiebt man die Daten so, dass ihr Mittelwert im Nullpunkt liegt. Hierfür berechnet man zuerst den Mittelwert

$$\bar{X} = \frac{1}{n}(X_1 + \dots + X_n) = (\bar{x}_1, \dots, \bar{x}_d)$$

und ermittelt mit diesem die verschobenen Datenpunkte

$$X_i^{\text{neu}} = (x_{i1}^{\text{neu}}, \dots, x_{id}^{\text{neu}}) \forall i = 1, \dots, n,$$

mit $x_{ij}^{\text{neu}} = x_{ij} - \bar{x}_j \forall i = 1, \dots, n, j = 1, \dots, d$. Schreibt man nun diese Datenpunkte in eine Matrix

$$A = \begin{pmatrix} x_{11}^{\text{neu}} & \dots & x_{n1}^{\text{neu}} \\ \vdots & \ddots & \vdots \\ x_{1d}^{\text{neu}} & \dots & x_{nd}^{\text{neu}} \end{pmatrix},$$

und berechnet die Kovarianzmatrix

$$C = \frac{1}{n-1}AA^T,$$

4 Approximation mit einfachen Objekten

lässt sich sofort sehen, dass die Diagonaleinträge der Kovarianzmatrix C die Varianzen der einzelnen Komponenten der Datenpunkte darstellen:

$$C = \frac{1}{n-1}AA^T = \frac{1}{n-1} \begin{pmatrix} x_{11}^{\text{neu}} & \dots & x_{n1}^{\text{neu}} \\ \vdots & \ddots & \vdots \\ x_{1d}^{\text{neu}} & \dots & x_{nd}^{\text{neu}} \end{pmatrix} \begin{pmatrix} x_{11}^{\text{neu}} & \dots & x_{1d}^{\text{neu}} \\ \vdots & \ddots & \vdots \\ x_{n1}^{\text{neu}} & \dots & x_{nd}^{\text{neu}} \end{pmatrix} = \frac{1}{n-1} \begin{pmatrix} c_{11} & \dots & c_{1d} \\ \vdots & \ddots & \vdots \\ c_{d1} & \dots & c_{dd} \end{pmatrix},$$

mit

$$c_{kk} = \frac{1}{n-1} \sum_{i=1}^n (x_{ki}^{\text{neu}})^2 = \frac{1}{n-1} \sum_{i=1}^n (x_{ki} - \bar{x}_i)^2 = \text{Var}_k$$

und der Varianz Var_k der k -ten Koordinaten.

Aufgrund der Symmetrie dieser Matrix, kann man Satz 2.4 anwenden und erhält eine Diagonalisierung der Kovarianzmatrix

$$C = Y\Lambda Y^{-1}$$

mit

$$\Lambda = \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_d \end{pmatrix}$$

eine Diagonalmatrix mit den Eigenwerten der Kovarianzmatrix und

$$Y = (\vec{v}_1, \dots, \vec{v}_d),$$

wobei \vec{v}_i den Eigenvektor zu dem Eigenwert λ_i darstellt. Die Eigenvektoren

$$\vec{v}_1, \dots, \vec{v}_d$$

werden auch die *Hauptkomponenten* genannt.

Betrachtet man nun die Spur der Kovarianzmatrix C , ist diese einerseits die Summe der Varianzen der einzelnen Dimensionen der Datenpunkte

$$\text{tr}(C) = \text{Var}_1 + \dots + \text{Var}_d =: S$$

und andererseits die Summe der Eigenwerte

$$\text{tr}(C) = \lambda_1 + \dots + \lambda_d.$$

Somit sieht man, dass der Eigenvektor \vec{v}_i einen Anteil

$$\frac{\lambda_i}{S}$$

zur Summe der Varianzen beiträgt.

4 Approximation mit einfachen Objekten

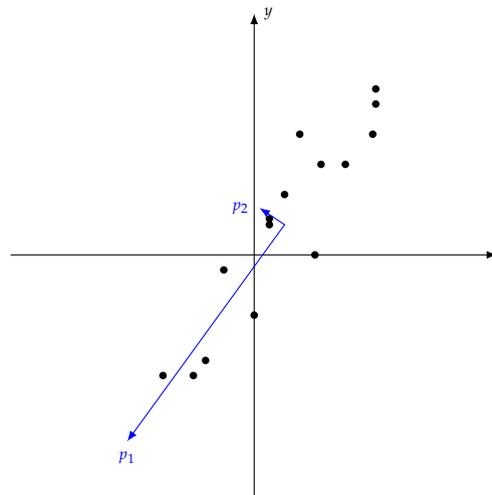


Abbildung 4.2: Die zwei Hauptkomponenten einer Datenmenge im \mathbb{R}^2

Seien nun o.B.d.A

$$\lambda_1 \geq \dots \geq \lambda_d \geq 0$$

geordnet in absteigender Reihenfolge und

$$\vec{v}_1, \dots, \vec{v}_d$$

die Eigenvektoren mit \vec{v}_i Eigenvektor zu Eigenwert λ_i , dann trägt mit den obigen Argumenten der Eigenwert λ_1 den größten Teil zur Gesamtvarianz S bei und somit ist der zugehörige Eigenvektor \vec{v}_1 die „signifikanteste“ Hauptkomponente für den gegebenen Datensatz, \vec{v}_2 die nächstwichtigste Hauptkomponente, und so weiter. Möchte man also einen d -dimensionalen Datensatz auf p Dimensionen reduzieren, wobei $p \leq d$, betrachtet man nur die p signifikantesten Hauptkomponenten [14].

Beispiel 4.1. Es sei $\{X_i\}_{1 \leq i \leq 100}$ ein 8-dimensionaler Datensatz mit Summe der Varianzen $S = 150$. Da es schwierig, wenn nicht gar unmöglich, ist, dies visuell darzustellen, versucht man mittels PCA die Dimensionen zu reduzieren. Als Eigenwerte bekommt man $\lambda_1 = 72$, $\lambda_2 = 40$, $\lambda_3 = 37$ und λ_4 bis λ_8 jeweils 0.2. Dies lässt darauf schließen, dass die Richtungen der drei Eigenvektoren \vec{v}_1 , \vec{v}_2 und \vec{v}_3 die bedeutendsten sind, und man durch Reduzierung auf diese drei Richtungen nicht besonders viel Information verliert. Somit kann man also einen 8-dimensionalen Datensatz auf drei Dimensionen reduzieren, die man nun auch visuell darstellen und analysieren kann.

Hier noch einmal eine Zusammenfassung der Hauptkomponentenanalyse, wenn man eine Menge von d -dimensionalen Daten $\{X_i\}_{1 \leq i \leq n}$ gegeben hat:

- Bestimmen der Kovarianzmatrix
- Berechnen der Eigenwertzerlegung der Kovarianzmatrix
- Wahl der „signifikantesten“ p Hauptkomponenten (Eigenvektoren) zum Reduzieren der Dimensionen

4.2.1 PCA zum Approximieren mit einfachen Objekten

In Abschnitt 4.1 wurde schon mit Hilfe der Hauptkomponentenanalyse eine Datenmenge mittels einer Geraden bestmöglichst im Sinne des geometrischen Abstandes approximiert, indem der Schwerpunkt und die signifikanteste Hauptkomponente verwendet wurde.

Um zu zeigen, wie man die Hauptkomponentenanalyse zum Approximieren von Objekten nutzen kann, betrachten wir folgende Lemmata [25]:

Lemma 4.2. Sei $\{X_i\}_{1 \leq i \leq n}$ eine Punktmenge im \mathbb{R}^m . Dann ist der Punkt, der die Zielfunktion $F(X_0) = \sum_{i=1}^n \|X_i - X_0\|^2$ minimiert, der Schwerpunkt der Datenpunkte. Für jeden anderen Punkt $\tilde{X}_0 \neq X_0$ gilt

$$F(\tilde{X}_0) > F(X_0).$$

Beweis. Verschiebt man die Punktmenge um einen Parameter $\vec{a} \in \mathbb{R}^m$, bekommt man eine Punktmenge $\{X_i^{\text{neu}}\}_{1 \leq i \leq n}$ mit

$$X_i^{\text{neu}} = X_i + \vec{a},$$

wodurch sich der Schwerpunkt der verschobenen Punktmenge ändert zu

$$X_0^{\text{neu}} = \frac{1}{n} \sum_{i=1}^n (X_i + \vec{a}) = \frac{1}{n} \sum_{i=1}^n X_i + n \frac{\vec{a}}{n} = X_0 + \vec{a}.$$

Bei Verschiebung der Punktmenge verschiebt sich somit der Schwerpunkt um den gleichen Parameter.

Betrachtet man o.B.d.A eine Punktmenge mit Schwerpunkt X_0 im Ursprung und einen weiteren beliebigen Punkt $\tilde{X}_0 \neq X_0$, folgt wegen $X_0 = \frac{1}{n} \sum_{i=1}^n X_i = 0$

$$\begin{aligned} F(\tilde{X}_0) - F(X_0) &= \sum_{i=1}^n \|X_i - \tilde{X}_0\|^2 - \sum_{i=1}^n \|X_i - X_0\|^2 \\ &= \sum_{i=1}^n \|X_i\|^2 - \sum_{i=1}^n 2\langle X_i, \tilde{X}_0 \rangle + \sum_{i=1}^n \|\tilde{X}_0\|^2 - \sum_{i=1}^n \|X_i\|^2 \\ &= -2\langle \sum_{i=1}^n X_i, \tilde{X}_0 \rangle + n\|\tilde{X}_0\|^2 \\ &= n\|\tilde{X}_0\|^2, \end{aligned}$$

wodurch die Behauptung folgt. □

Lemma 4.3. Sei $\{X_i\}_{1 \leq i \leq n}$ eine Punktmenge und \vec{a} ein Einheitsvektor im \mathbb{R}^m . Dann führt die Gerade

$$X_0 + u \cdot \vec{a} \text{ mit } u \in \mathbb{R},$$

4 Approximation mit einfachen Objekten

die die Quadratsumme der orthogonalen Abstände zur Punktmenge minimiert, durch den Schwerpunkt dieser Menge. Jede andere Gerade mit Richtung \vec{a} , die nicht durch den Schwerpunkt führt, hat eine strikt größere Quadratsumme.

Beweis. Man betrachte eine Gerade mit Richtung \vec{a} , die nicht durch den Schwerpunkt der Punktmenge führt und die Abstände zu dieser minimiert, und konstruiert eine Ebene, die normal zur Geraden steht und durch den Schwerpunkt \bar{X} geht. Sei $X^* \neq \bar{X}$ der Schnittpunkt der Geraden mit der Ebene. Die Zielfunktion dieser Problemstellung lässt sich mittels Satz von Pythagoras formulieren als

$$F(X^*) = \sum_{i=1}^n \left(\|X_i - X^*\|^2 - (d^{\text{plane}}(X_i, X^*))^2 \right).$$

Für die Gerade mit Richtung \vec{a} , die durch den Schwerpunkt geht, gilt dieselbe Zielfunktion:

$$F(\bar{X}) = \sum_{i=1}^n \left(\|X_i - \bar{X}\|^2 - (d^{\text{plane}}(X_i, \bar{X}))^2 \right).$$

Da die Ebene so konstruiert wurde, dass sie sowohl den Schwerpunkt \bar{X} , als auch den Schnittpunkt X^* enthält, folgt

$$d^{\text{plane}}(X_i, \bar{X}) = d^{\text{plane}}(X_i, X^*)$$

und die Differenz der Zielfunktionen reduziert sich auf

$$F(X^*) - F(\bar{X}) = \sum_{i=1}^n (\|X_i - X^*\|^2) - \sum_{i=1}^n (\|X_i - \bar{X}\|^2).$$

Da wir aus dem vorherigen Lemma wissen, dass

$$\sum_{i=1}^n (\|X_i - X^*\|^2) > \sum_{i=1}^n (\|X_i - \bar{X}\|^2)$$

und somit

$$F(X^*) - F(\bar{X}) > 0,$$

besitzt die Gerade mit Richtung \vec{a} , die nicht durch den Schwerpunkt geht, eine größere Abstands-Quadratsumme.

Somit ist die Gerade mit Richtung \vec{a} , die durch den Schwerpunkt geht, der eindeutige Minimierer der Quadratsumme der orthogonalen Distanzen zur Punktmenge. \square

Satz 4.4. Sei $\{X_i\}_{1 \leq i \leq n} \in \mathbb{R}^m$ eine Punktmenge. Dann enthält die Gerade, die die orthogonalen Distanzen zu der Punktmenge im Sinne der kleinsten Fehlerquadrate minimiert, den Schwerpunkt der Punktmenge und zeigt in Richtung des Eigenvektors mit dem größten Eigenwert der Kovarianzmatrix der in den Ursprung verschobenen Datenmatrix

$$A = \begin{pmatrix} (x_{11} - x_{01}) & \dots & (x_{n1} - x_{01}) \\ \vdots & \ddots & \vdots \\ (x_{1m} - x_{0m}) & \dots & (x_{nm} - x_{0m}) \end{pmatrix},$$

4 Approximation mit einfachen Objekten

mit Schwerpunkt X_0 und $X_i = \begin{pmatrix} x_{i1} \\ \vdots \\ x_{im} \end{pmatrix} \forall i = 0, \dots, n.$

Beweis. Dass die gesuchte Gerade den Schwerpunkt der Punktmenge enthalten muss, wurde schon mit Lemma 4.2 gezeigt.

Sei o.B.d.A. $\{X_i\}_{1 \leq i \leq n} \in \mathbb{R}^m$ eine Menge von Punkten mit dem Schwerpunkt im Ursprung. Gesucht wird nun eine Richtung \vec{v} mit $\|\vec{v}\| = 1$, die die Quadratsumme der orthogonalen Abstände zur Punktmenge

$$\sum_{i=1}^n d_i^2 = \sum_{i=1}^n \|X_i - \vec{v}\vec{v}^T X_i\|^2$$

minimiert, wobei $\vec{v}\vec{v}^T X_i$ den Punkt X_i projiziert auf die durch den Ursprung gehende Gerade mit Richtung \vec{v} darstellt.

Es gilt nun

$$\begin{aligned} d_i^2 &= \|X_i - \vec{v}\vec{v}^T X_i\|^2 \\ &= (X_i - \vec{v}\vec{v}^T X_i)^T (X_i - \vec{v}\vec{v}^T X_i) \\ &= X_i^T X_i - X_i^T \vec{v}\vec{v}^T X_i - X_i^T \vec{v}\vec{v}^T X_i + X_i^T \vec{v}\vec{v}^T \vec{v}\vec{v}^T X_i \\ &= X_i^T X_i - \|X_i^T \vec{v}\|^2. \end{aligned}$$

Bilden wir den Durchschnitt über alle Punkte, so folgt schließlich

$$\begin{aligned} \frac{1}{n} \sum_{i=1}^n d_i^2 &= \frac{1}{n} \sum_{i=1}^n X_i^T X_i - \frac{1}{n} \sum_{i=1}^n \|X_i^T \vec{v}\|^2 \\ &= \frac{1}{n} X^T X - \frac{1}{n} \|X^T \vec{v}\|^2, \end{aligned}$$

wobei X eine $(m \times n)$ -Matrix mit dem Datenpunkt X_i in Spalte i ist. Da der erste Term auf der rechten Seite fix ist, folgt also, dass das Minimieren der Distanzen dem Maximieren des zweiten Terms der Gleichung entspricht.

Wir suchen also einen Einheitsvektor \vec{v} , der V maximiert. Betrachtet man V näher, lässt sich diese Matrix auch darstellen als

$$\begin{aligned} \frac{1}{n} \|X^T \vec{v}\|^2 &= \frac{1}{n} (X^T \vec{v})^T (X^T \vec{v}) \\ &= \frac{1}{n} (\vec{v}^T X X^T \vec{v}) \\ &= \vec{v}^T \left(\frac{1}{n} X X^T \right) \vec{v} \\ &= \vec{v}^T C \vec{v}, \end{aligned}$$

4 Approximation mit einfachen Objekten

mit der Kovarianzmatrix C der Datenmatrix X . Dies ergibt somit folgendes Optimierungsproblem

$$\max V = \max_{\|\vec{v}\|=1} \vec{v}^T C \vec{v}$$

welches man mit dem Verfahren der Lagrange-Multiplikatoren lösen kann.

Für dieses Verfahren bildet man aus dem gegebenen Optimierungsproblem die Lagrange-Funktion

$$L = \vec{v}^T C \vec{v} - \lambda(\vec{v}^T \vec{v} - 1),$$

mit dem Lagrange-Multiplikator λ . Ableiten nach \vec{v} und Nullsetzen ergibt

$$\frac{\partial L}{\partial \vec{v}} = 2\vec{v}^T C - 2\lambda\vec{v}^T = 0,$$

woraus sofort

$$\vec{v}^T C = \lambda\vec{v}^T$$

folgt. Auf Grund der Symmetrie der Kovarianzmatrix ergibt Transponieren der beiden Seiten die Gleichung

$$C\vec{v} = \lambda\vec{v},$$

womit man beim Eigenwertproblem angelangt ist.

Betrachtet man nun noch die alternative Lagrange-Funktion

$$L^* = \mu\vec{v}^T C \vec{v} - (\vec{v}^T \vec{v} - 1),$$

mit dem Lagrange-Multiplikator μ , und bildet seine Ableitung

$$\frac{\partial L^*}{\partial \vec{v}} = 2\mu\vec{v}^T C - 2\vec{v}^T = 0,$$

muss noch gezeigt werden, dass mit $\mu = 0$ diese Gleichung keine Lösung für das Maximierungsproblem darstellen kann. Dies ist jedoch gegeben, da für $\mu = 0$

$$2\mu\vec{v}^T C - 2\vec{v}^T = 0 \rightarrow \vec{v}^T = 0$$

gilt, und \vec{v} somit nicht die Nebenbedingung $\|\vec{v}\| = 1$ erfüllt.

Um die Frage zu beantworten, welcher Eigenvektor die Lösung dieses Maximierungsproblems darstellt, betrachten wir folgende Gleichung mit dem Eigenvektor \vec{v}_k und dem zugehörigen Eigenwert λ_k :

$$\frac{1}{n} \|X^T \vec{v}_k\|^2 = \vec{v}_k^T C \vec{v}_k = \vec{v}_k^T \lambda_k \vec{v}_k = \lambda_k \vec{v}_k^T \vec{v}_k = \lambda_k,$$

womit man sieht, dass das Einsetzen eines Eigenvektors in die Gleichung den zugehörigen Eigenwert ergibt.

Hat man also Eigenvektoren $\vec{v}_1, \dots, \vec{v}_n$ mit zugehörigen Eigenwerten $\lambda_1 \geq \dots \geq \lambda_n$ gegeben, muss der Eigenvektor \vec{v}_1 mit dem zugehörigen größten Eigenwert λ_1 als erste Hauptkomponente herangezogen werden, um die Varianz zu maximieren. \square

4 Approximation mit einfachen Objekten

Satz 4.5. Seien Punkte $\{X_i\}_{1 \leq i \leq n} \in \mathbb{R}^m$ gegeben und U ein k -dimensionaler affiner Unterraum, der die quadrierten orthogonalen Distanzen zur Punktmenge

$$F(U) = \sum_{i=1}^n d(X_i, U)^2$$

minimiert, $\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i$ der Schwerpunkt der Punktmenge und $\lambda_1 \geq \dots \geq \lambda_n$ die Eigenwerte der Kovarianzmatrix $C = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})(X_i - \bar{X})^T$ mit den zugehörigen Eigenvektoren $\vec{v}_1, \dots, \vec{v}_n$.

Dann gilt:

1. Der Unterraum U enthält den Schwerpunkt \bar{X} der Punktmenge.
2. Der Unterraum U wird von den Eigenvektoren $\vec{v}_1, \dots, \vec{v}_k$ aufgespannt.

Bevor wir uns dem Beweis dieses Satzes widmen können, benötigen wir eine weitere Eigenschaft der Eigenwerte, die auch als Courantsches Minimax-Prinzip bekannt ist [5]:

Satz 4.6. Sei $A \in \mathbb{R}^{n \times n}$ eine symmetrische Matrix mit den Eigenwerten

$$\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n,$$

dann gilt

$$\lambda_k = \min_{U_k \in \mathcal{U}_k} \max_{\vec{x} \in U_k, \|\vec{x}\|=1} \vec{x}^T A \vec{x},$$

mit $\mathcal{U}_k = \{U_k \subset \mathbb{R}^n \mid U_k \text{ Unterraum mit } \dim(U_k) = n - k + 1\}$.

Beweis. Die symmetrische Matrix A besitzt eine Eigenwertzerlegung in ein Orthonormalsystem $V = (\vec{v}_1, \dots, \vec{v}_n)$ von Eigenvektoren zu den Eigenwerten $\lambda_1, \dots, \lambda_n$.

Durch die Spektraldarstellung ergibt sich

$$A = \sum_{i=1}^n \lambda_i \vec{v}_i \vec{v}_i^T.$$

Betrachtet man für ein festes k den Unterraum

$$V_k = \text{span}\{\vec{v}_1, \dots, \vec{v}_k\}$$

und wählt $U_k \in \mathcal{U}_k$ beliebig, existiert wegen

$$\dim(V_k \cap U_k) = \dim V_k + \dim U_k - \dim(U_k + V_k) = n + 1 - \dim(U_k + V_k) \geq 1$$

ein Einheitsvektor $\vec{x}^* \in V_k \cap U_k$, der sich eindeutig darstellen lässt als

$$\vec{x}^* = \sum_{i=1}^k \alpha_i \vec{v}_i$$

4 Approximation mit einfachen Objekten

mit $\sum_{i=1}^k \alpha_j^2 = 1$, da

$$1 = \|\vec{x}^*\|^2 = \sum_{i=1}^k \|\alpha_j \vec{v}_j\|^2 = \sum_{i=1}^k \alpha_j^2 \vec{v}_j^T \vec{v}_j = \sum_{i=1}^k \alpha_j^2.$$

Da $\vec{x}^* \in V_k$ und somit $\vec{v}_i^T \vec{x}^* = 0 \forall i = k+1, \dots, n$, folgt

$$\vec{x}^{*T} A \vec{x}^* = \sum_{i=1}^n \lambda_i \|\vec{v}_i^T \vec{x}^*\|^2 = \sum_{i=1}^k \lambda_i \|\vec{v}_i^T \vec{x}^*\|^2 = \sum_{i=1}^k \lambda_i \alpha_i^2.$$

Hieraus erhält man

$$\max_{\vec{x} \in U_k, \|\vec{x}\|=1} \vec{x}^T A \vec{x} \geq \vec{x}^{*T} A \vec{x}^* = \sum_{i=1}^k \lambda_i \alpha_i^2 \geq \lambda_k \sum_{i=1}^k \alpha_i^2 = \lambda_k,$$

und da U_k beliebig gewählt wurde, folgt

$$\min_{U_k \in \mathcal{U}_k} \max_{\vec{x} \in U_k, \|\vec{x}\|=1} \vec{x}^T A \vec{x} \geq \lambda_k.$$

Wählt man nun den speziellen Unterraum

$$W_k = \text{span}\{\vec{v}_k, \dots, \vec{v}_n\},$$

gilt $\dim(W_k) = n - k + 1$ und jeder Einheitsvektor $\vec{x}^* \in W_k$ lässt sich eindeutig darstellen als

$$\vec{x}^* = \sum_{i=k}^n \beta_j \vec{v}_j$$

mit $\sum_{i=1}^k \beta_j^2 = 1$ wie zuvor. Da $W_k \in \mathcal{U}_k$ folgt

$$\min_{U_k \in \mathcal{U}_k} \max_{\vec{x} \in U_k, \|\vec{x}\|=1} \vec{x}^T A \vec{x} \leq \max_{\vec{x} \in W_k, \|\vec{x}\|=1} \vec{x}^T A \vec{x} = \sum_{i=k}^n \lambda_i \beta_i^2 \leq \lambda_k \sum_{i=k}^n \beta_i^2 = \lambda_k,$$

womit die Behauptung gilt. □

Korollar 4.7. Sei $A \in \mathbb{R}^{n \times n}$ eine symmetrische Matrix mit den Eigenwerten

$$\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n.$$

Dann gilt für die Summe von k Hauptdiagonalelementen der Matrix A , o.B.d.A von a_{11}, \dots, a_{kk} , dass

$$\lambda_1 + \dots + \lambda_k \leq a_{11} + \dots + a_{kk}.$$

4 Approximation mit einfachen Objekten

Beweis. Nimmt man das wie in Satz 4.6 definierte Minimum nur über die Unterräume von $\text{span}\{\vec{e}_1, \dots, \vec{e}_k\}$, wobei $\vec{e}_i = (\delta_i)_j$, erhält man die Eigenwerte

$$\lambda'_1 \leq \dots \leq \lambda'_k$$

der linken oberen $(k \times k)$ -Teilmatrix A' von A .

Auf Grund von Satz 4.6 gilt $\lambda'_i \geq \lambda_i$. Dies gilt, da für die Berechnung von λ'_i das Minimum über eine kleinere Menge von Unterräumen angenommen wird. Nun folgt direkt

$$\text{tr}(A') = \lambda'_1 + \dots + \lambda'_k \geq \lambda_1 + \dots + \lambda_k.$$

□

Nun zum Beweis von Satz 4.5:

Beweis. Sei U ein k -dimensionaler affiner Unterraum, der den Schwerpunkt \bar{X} der gegebenen Punktmenge enthält, wobei dieser im Koordinatensystem im Ursprung liegt, d.h. $\bar{X} = 0$. Mit einer Verschiebung U' des Unterraumes U um einen Parameter $\vec{a} \in U^\perp$ erhält man

$$\begin{aligned} F(U') &= \sum_{i=1}^n d(X_i - \vec{a}, U)^2 \\ &= \sum_{i=1}^n \|\tilde{X}_i - \vec{a}\|^2 \\ &= \sum_{i=1}^n \|\tilde{X}_i\|^2 - 2 \left\langle \sum_{i=1}^n \tilde{X}_i, \vec{a} \right\rangle + \sum_{i=1}^n \|\vec{a}\|^2 \\ &= \sum_{i=1}^n d(X_i, U)^2 + \|\vec{a}\|^2 \\ &= F(U) + \|\vec{a}\|^2, \end{aligned}$$

wobei \tilde{X}_i die Projektion des Punktes X_i auf den Orthogonalraum U^\perp darstellt. Somit gilt, dass $F(U) \leq F(U')$, wobei Gleichheit nur für $\vec{a} = 0$ erfüllt ist. Ein Minimum U muss also den Schwerpunkt enthalten.

Für den zweiten Punkt betrachten wir eine Orthonormalbasis $B = (b_1, \dots, b_m)$ und setzen $L_B = \text{span}\{b_{m-k+1}, \dots, b_m\}$. Betrachtet man die Zielfunktion

$$F(L_B) = \sum_{i=1}^n d(X_i, L_B)^2 = \sum_{i=1}^n \sum_{j=1}^{m-k} \langle X_i, b_j \rangle^2,$$

entspricht dies der Summe der $m - k$ Hauptdiagonalelemente der Matrix $B^T C B$, da

$$(B^T C B)_{k,k} = \left(\sum_{i=1}^n B^T X_i X_i^T B \right)_{k,k} = \sum_{i=1}^n \langle b_k, X_i \rangle \langle X_i, b_k \rangle.$$

4 Approximation mit einfachen Objekten

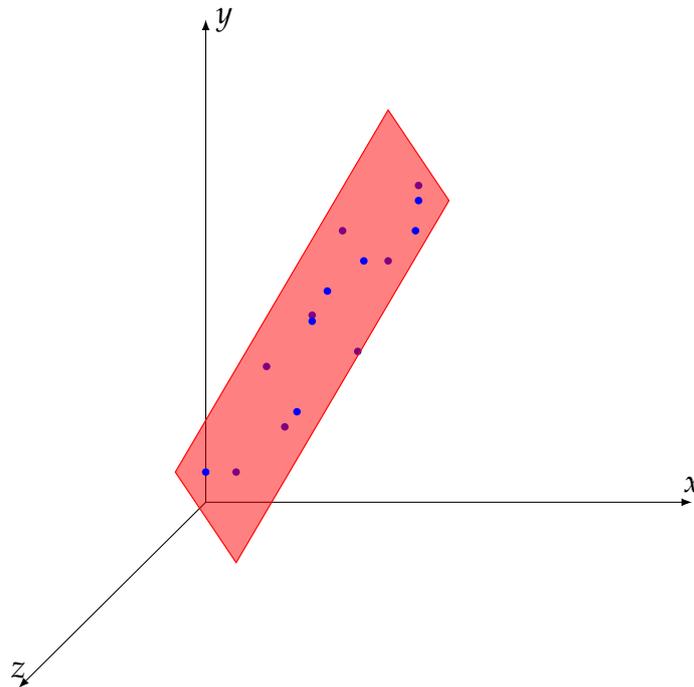


Abbildung 4.3: Approximieren einer Datenmenge mittels einer Ebene im \mathbb{R}^3

Auf Grund des Korollars 4.2.1 ist die Summe der kleinsten $m - k$ Eigenwerte eine untere Schranke für diese Summe. Wählt man die Matrix B als Eigenbasis von C , aufsteigend sortiert nach den Eigenwerten, ergibt $B^T C B$ eine Diagonalmatrix, womit die Schranke angenommen wird und der Unterraum L_B die Zielfunktion $F(L_B)$ minimiert. \square

Dieses Prinzip lässt sich nun folgendermaßen auf das Approximieren mittels einer Ebene anwenden:

Für diese benötigt man nun zwei Richtungen und auf Grund des vorigen Satzes sind dies die zwei signifikantesten Hauptkomponenten. Somit hat man als Lösung die Ebene in Parameterdarstellung:

$$\begin{pmatrix} x_1 \\ \vdots \\ x_d \end{pmatrix} = \bar{X} + u \cdot \vec{v}_1 + v \cdot \vec{v}_2,$$

wobei \bar{X} den Mittelwert der Datenpunkte und \vec{v}_1 und \vec{v}_2 die signifikantesten Hauptkomponenten darstellen (siehe Abbildung 4.3, hier wurden der Anschaulichkeit halber Datenpunkte im \mathbb{R}^3 gewählt).

Allgemein ergibt sich folgender Algorithmus zum Approximieren einer n -dimensionalen Punktmenge mit einem k -dimensionalen affinen Unterraum:

4 Approximation mit einfachen Objekten

1. Berechnen des Schwerpunktes \bar{X} der Punktmenge
2. Aufstellen der Kovarianzmatrix
3. Berechnen der Eigenwerte $\lambda_1 \geq \dots \geq \lambda_n$ und der zugehörigen Eigenvektoren $\vec{v}_1, \dots, \vec{v}_n$ der Kovarianzmatrix
4. Aufstellen des Unterraumes mit Hilfe des Schwerpunktes \bar{X} und der Eigenvektoren $\vec{v}_1, \dots, \vec{v}_k$

4.2.2 PCA und Singulärwertzerlegung

In Abschnitt 4.2.1 wurde gezeigt, dass die Hauptkomponentenanalyse die Eigenwertzerlegung der Kovarianzmatrix der gegebenen Punktmenge zum Approximieren von affinen Unterräumen verwendet.

Nun gibt es jedoch noch eine zweite Variante, die statt der Eigenwertzerlegung der Kovarianzmatrix die Singulärwertzerlegung der zentrierten Punktmenge verwendet.

Diese Variante wird folgendermaßen angewendet:

1. Zentrieren der gegebenen Punktmenge
2. Berechnen der Singulärwertzerlegung der zentrierten Punktmenge
3. Aufstellen des Unterraumes mit Hilfe des Schwerpunktes \bar{X} und der Singulärvektoren $\vec{v}_1, \dots, \vec{v}_k$, die zu den k -größten Singulärwerten $\sigma_1 \geq \dots \geq \sigma_k$ gehören.

Um die Korrektheit dieser Variante zu zeigen, sei $A = (X_1, \dots, X_n)$ eine $d \times n$ -Matrix mit n Datenpunkten im \mathbb{R}^d und o.B.d.A der Mittelwert der gegebenen Daten gleich Null (ansonsten den Mittelwert berechnen und von den Daten abziehen). Laut Satz 2.11 besitzt diese Matrix eine Singulärwertzerlegung

$$A = U\Sigma V^T,$$

wobei $U \in \mathbb{R}^{d \times d}$ die linken Singulärvektoren, $V \in \mathbb{R}^{n \times n}$ die rechten Singulärvektoren, und $\Sigma = (s_{i,j}) \in \mathbb{R}^{m \times n}$ die Singulärwerte $s_{1,1} \geq s_{2,2} \geq \dots$ enthält. Betrachtet man nun die Kovarianzmatrix C , folgt

$$\begin{aligned}(n-1)C &= A^T A \\ &= (U\Sigma V^T)^T (U\Sigma V^T) \\ &= V\Sigma U^T U\Sigma V^T \\ &= V\Sigma^2 V^T,\end{aligned}$$

da U eine orthogonale Matrix ist.

Diese Gleichung ergibt also eine Eigenwertentwicklung der Kovarianzmatrix mittels der Singulärwertzerlegung der Datenmatrix, wobei für die Singulärwertzerlegung der Datenmatrix $A = U\Sigma V^T$, die Matrix V die Eigenvektoren und die Matrix Σ die Wurzeln der Eigenwerte der Kovarianzmatrix beinhalten.

4 Approximation mit einfachen Objekten

Ein wichtiger Schritt in beiden Varianten ist das Zentrieren der Datenpunkte. Wir betrachten in einem Beispiel, was geschieht, wenn man die Hauptkomponentenanalyse mit der Singulärwertzerlegung durchführt, ohne die Daten zuvor zu zentrieren. Hierfür betrachten wir eine Menge von Punkten im \mathbb{R}^2 , ersichtlich in Tabelle 4.3.

Tabelle 4.3: Punktmenge zum Vergleich der Approximation mit und ohne Zentrieren

| | p1 | p2 | p3 | p4 | p5 | p6 | p7 | p8 | p9 | p10 | p11 | p12 | p13 | p14 | p15 |
|---|-----|-----|-----|-----|----|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| x | 3.9 | 4.3 | 3.5 | 4.7 | 4 | 4 | 3.9 | 4.1 | 3.9 | 4 | 4.1 | 4 | 3.8 | 3.9 | 4.2 |
| y | 8.1 | 3.3 | 5 | 5.3 | 4 | 6 | 7.5 | 6.1 | 4.1 | 5 | 5.2 | 7.8 | 7.3 | 5.1 | 2 |

Mit Zentrieren der Daten bekommt man als Ergebnis der Singulärwertzerlegung der Datenmatrix \bar{X} mit den zentrierten Punkten

$$\bar{X}_{\text{centered}} = \begin{pmatrix} -0.0462969 & -0.998928 \\ 0.998928 & -0.0462969 \end{pmatrix} \begin{pmatrix} 6.48427 & 0 & 0 & \dots & 0 \\ 0 & 0.935818 & 0 & \dots & 0 \end{pmatrix} \bar{V}^T,$$

mit \bar{V} eine (15×15) -Matrix, die die rechten Singulärvektoren beinhaltet.

Führen wir nun eine Singulärwertzerlegung der Datenmatrix X durch, ohne die Punkte zuvor zu zentrieren, bekommt man folgendes Ergebnis:

$$\bar{X} = \begin{pmatrix} -0.56991 & -0.821707 \\ -0.821707 & 0.56991 \end{pmatrix} \begin{pmatrix} 26.7363 & 0 & 0 & \dots & 0 \\ 0 & 4.07185 & 0 & \dots & 0 \end{pmatrix} V^T,$$

wobei V wieder eine (15×15) -Matrix darstellt, die die rechten Singulärvektoren beinhaltet.

Betrachtet man nun die Geraden, die aus Eigenvektoren mit dem jeweils größten Eigenwert dieser zwei Zerlegungen gebildet werden, bekommt man das in Abbildung 4.4 ersichtliche Ergebnis. Hier sieht man, dass die Gerade, die mit der Methode ohne Zentrieren entsteht, nicht der Geraden entspricht, die die orthogonalen Abstände zu der Punktmenge minimiert, da der Eigenvektor mit dem größten Eigenwert im Fall ohne Zentrieren der Datenpunkte nicht mehr in Richtung der größten Varianz der Daten zeigt. [7]

4.2.3 Robuste Hauptkomponentenanalyse

Ein großes Problem der Hauptkomponentenanalyse stellen Datensätze dar, die Ausreißer enthalten.

Verwendet man die Hauptkomponentenanalyse zum Einpassen einer Geraden in eine Menge von Punkten im \mathbb{R}^2 , bekommt man das in Abbildung 4.5 links ersichtliche Ergebnis. Fügt man dieser Punktmenge einen Ausreißer hinzu und approximiert sie

4 Approximation mit einfachen Objekten

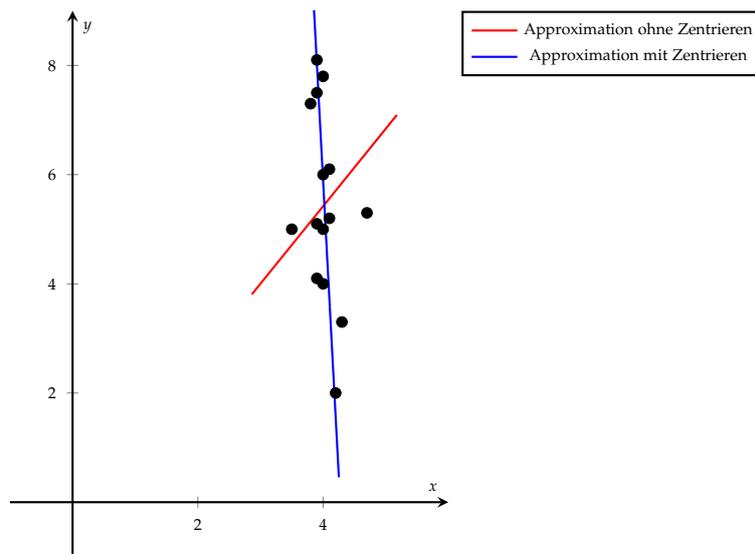


Abbildung 4.4: Ergebnis der Approximation mittels einer Geraden mit und ohne Zentrieren der Punktmenge

mit einer Geraden mittels der gleichen Methode (siehe Abbildung 4.5 rechts) ist eine erhebliche Abweichung ersichtlich.

Dies ist nun nicht das gewünschte Ergebnis, da eine Approximation, wenn möglich, nicht so stark von Ausreißern beeinflusst werden sollte. Um ein Verfahren zu entwickeln, das nicht so empfindlich auf Ausreißer reagiert, wurde die robuste Hauptkomponentenanalyse (engl. „robust principle component analysis“ oder kurz „robust PCA“) eingeführt.

Als robuste Hauptkomponentenanalyse versteht man verschiedene Verfahren, die versuchen, die Hauptkomponentenanalyse für Punktmengen mit Ausreißern zu optimieren.

Es gibt unterschiedliche Herangehensweisen, PCA robuster zu gestalten, und im Folgenden werden zwei dieser Methoden vorgestellt:

- PCA mit Gewichtsiterationen und
- Zentrieren der Daten mittels $\| \cdot \|_1$ -Norm. [15]

PCA mit Gewichtsiterationen

Die erste hier betrachtete Möglichkeit, die Hauptkomponentenanalyse robuster gegenüber Ausreißern zu machen, ist die Hauptkomponentenanalyse mit Gewichtsiterationen.

4 Approximation mit einfachen Objekten

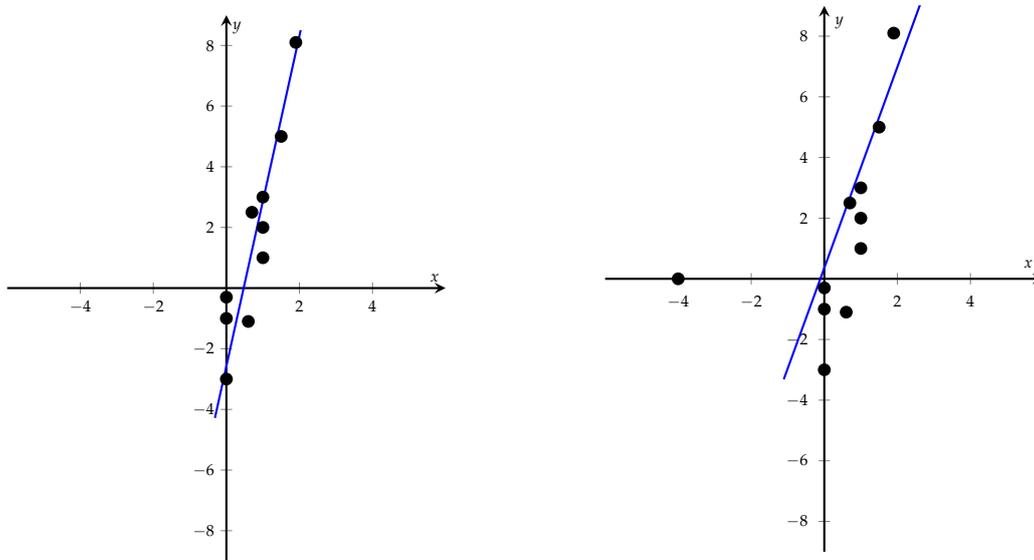


Abbildung 4.5: Approximieren von Datenpunkten ohne (links) und mit Ausreißern (rechts) mittels PCA

Für ein gegebenes Objekt O mit Objektparametern \vec{a} , welches die Punktmenge $\{X_i\}_{1 \leq i \leq n} \in \mathbb{R}^m$ approximiert, betrachtet man die zu minimierende Zielfunktion

$$\sigma(\vec{a})^2 = \sum_{i=1}^n \omega_i d(X_i, O(\vec{a}))^2,$$

wobei $d(X_i, O(\vec{a}))$ die Distanz vom Punkt X_i zum Objekt O mit Objektparametern \vec{a} und ω_i das Gewicht vom Punkt X_i darstellt.

Jeder gegebene Punkt in der Datenmenge bekommt also ein Gewicht, das seine Relevanz für die Zielfunktion angibt.

Man versucht nun Gewichte zu definieren, die Ausreißern ein sehr niedriges Gewicht (optimalerweise ein Gewicht von 0) zuweisen, wobei Datenpunkte, die für die Approximation relevant sind, hoch gewichtet werden. Um dies zu erreichen, betrachten wir folgende Vorgehensweise für ein gegebenes Objekt O und eine Punktmenge $\{X_i\}_{1 \leq i \leq n}$:

1. Setze das zu Punkt X_i gehörende Gewicht ω_i auf den Wert 1 $\forall i$ und führe die Approximationsaufgabe durch.
2. Berechne die Distanzen $d_i = d(X_i, O)$ für alle Punkte der gegebenen Menge.
3. Berechne die neuen Gewichte

$$\omega_i = \frac{1}{1 + \left(\frac{d_i}{\alpha}\right)^\beta}.$$

4. Führe die Approximationsaufgabe mit den neuen Gewichten durch.

4 Approximation mit einfachen Objekten

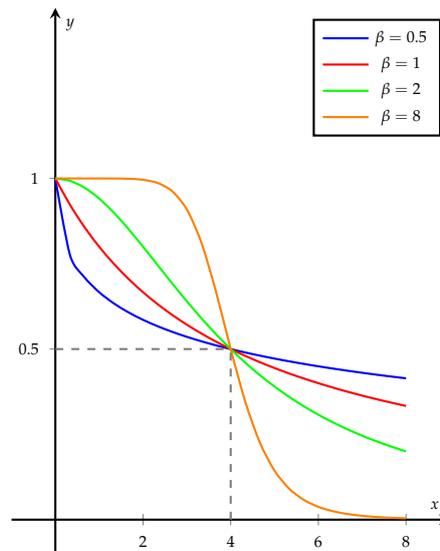


Abbildung 4.6: Gewichtsfunktion $y = \frac{1}{1+(\frac{x}{4})^\beta}$ für verschiedene Werte für β

5. Wiederhole Schritt 2 bis 4, bis gewisse Abbruchkriterien erfüllt sind.

Sehen wir uns zuerst Schritt 3 genauer an:

Wir haben die Funktion

$$y = \frac{1}{1 + (\frac{x}{\alpha})^\beta}$$

für fixe Werte von α und β .

Die x-Werte der Funktion entsprechen den gegebenen Distanzen des aktuellen Schritts und die y-Werte den zu berechnenden Gewichten. Nun stellt sich noch die Frage, wie die Parameter α und β gewählt werden sollten.

An der Gewichtsfunktion sieht man, dass je nach Wahl von α , Datenpunkte mit einer Distanz $d_i \geq \alpha$ ein Gewicht $\omega_i \leq \frac{1}{2}$ bekommen, da

$$\frac{1}{2} \geq \omega_i = \frac{1}{1 + (\frac{d_i}{\alpha})^\beta} \Leftrightarrow d_i \geq \alpha.$$

Einen guten Wert für α stellt zum Beispiel der Median der aktuellen Distanzen dar.

Die Wahl des Parameters β beeinflusst die Krümmung der Gewichtsfunktion. In Abbildung 4.6 sieht man den Unterschied der Gewichtsfunktionen für verschiedene Werte von β bei einem fixen Wert von $\alpha = 4$. Je größer der Parameter β gewählt wird, desto höher ist das Gewicht für Datenpunkte mit Distanzen $d_i < \alpha$ und niedriger für Datenpunkte mit Distanzen $d_i > \alpha$.

4 Approximation mit einfachen Objekten

In Schritt 4 werden die zuvor berechneten Gewichte für die Hauptkomponentenanalyse benutzt.

Hierfür betrachtet man den gewichteten Schwerpunkt der Punktmenge

$$\bar{X} = \frac{\sum_{i=1}^n \omega_i X_i}{\sum_{i=1}^n \omega_i}.$$

Setzt man alle Gewichte auf 1, wie in Schritt 1 des Algorithmus, bekommt man wieder den ungewichteten Schwerpunkt

$$S = \frac{\sum_{i=1}^n X_i}{n}.$$

Für den Varianzmaximierungsschritt betrachtet man die gewichtete Kovarianzmatrix $C = (c_{kl})_{1 \leq k, l \leq m}$ mit

$$c_{kl} = \frac{1}{\sum_{i=1}^n \omega_i} \sum_{i=1}^n \omega_i (X_{i,k} - \bar{X})(X_{i,l} - \bar{X})^T$$

und den Datenpunkten $\{X_i\}_{1 \leq i \leq n}$ mit $X_i = \begin{pmatrix} X_{i,1} \\ \vdots \\ X_{i,m} \end{pmatrix}$.

Im Fall, dass alle Gewichte gleich 1 sind, ergibt dies wieder die gewöhnliche Kovarianzmatrix.

Nun noch zu den Abbruchkriterien: Ein typisches Kriterium zum Abbruch solcher iterativen Verfahren ist, wenn sich die Zielfunktion nur mehr minimal ändert, das heißt

$$\sigma_k^2 - \sigma_{k+1}^2 < \epsilon,$$

wobei σ_i das Ergebnis der Zielfunktion im Iterationsschritt i darstellt und $\epsilon > 0$ fix gewählt wird.

Um den Abbruch des Algorithmus zu garantieren, wird meist auch eine maximale Iterationsschrittzahl festgelegt. Bei dieser Art von Gewichtsiteration ist diese Schranke sehr klein. Die maximale Schrittzahl kann zum Beispiel auf einen Wert kleiner 10 festgesetzt werden.

Wendet man die Hauptkomponentenanalyse mit Gewichtsiterationen auf das schon in Abbildung 4.5 gezeigte Beispiel von Datenpunkten mit einem Ausreißer an, wobei als Wert für α in der Gewichtsfunktion

$$\omega_i = \frac{1}{1 + \left(\frac{d_i}{\alpha}\right)^\beta}$$

der Median der in Schritt 2 berechneten Distanzen d_i und für β ein Wert von 10 verwendet wird, sieht man schon nach einer Iteration einen Unterschied in der Approximation (siehe Abbildung 4.7).

4 Approximation mit einfachen Objekten

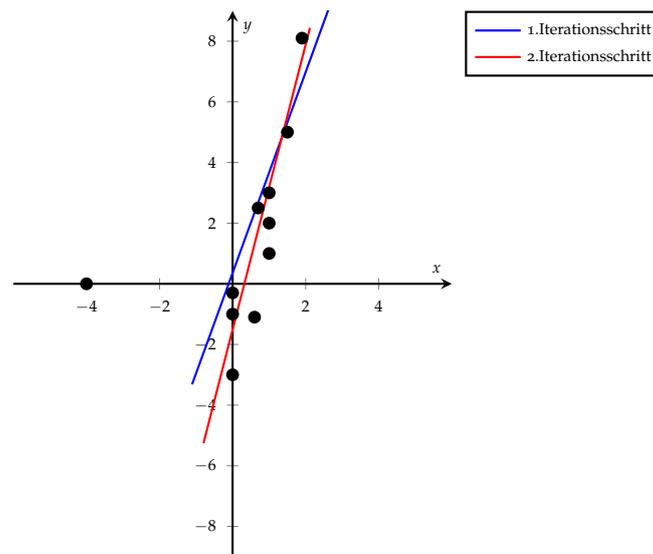


Abbildung 4.7: PCA mit Gewichtsiteration zum Approximieren einer Geraden in Datenpunkte mit Ausreißern

Die Gerade des ersten Iterationsschrittes entspricht der Geraden, die mit der Hauptkomponentenanalyse mit Gewichten $\omega_i = 1 \forall i$ approximiert wird. Schon im 2. Iterationsschritt stimmt die Gerade fast mit jener Geraden überein, die die Hauptkomponentenanalyse auch ohne Ausreißer ergeben hätte.

Hier wurde auf Grund der Verteilung der Punkte und der großen Distanz des Ausreißers ein großer Wert für den Parameter β in der Gewichtsfunktion gewählt, was dazu führt, dass dieser Datenpunkt schon nach einer Iteration ein Gewicht von fast 0 erhält.

Zentrieren der Daten mittels $\|\cdot\|_1$ -Norm

Im Gegensatz zum ursprünglichen Verfahren, das die euklidische Norm zum Zentrieren der Daten verwendet, wird bei dieser Variante der robusten PCA die $\|\cdot\|_1$ -Norm verwendet.

Betrachtet man eine eindimensionale Datenmenge $\{X_i\}_{1 \leq i \leq n}$, wird der Mittelwert berechnet durch

$$\bar{X} = \arg \min_{\bar{X} \in \mathbb{R}} \sum_{i=1}^n (X_i - \bar{X})^2.$$

Hier betrachten wir nun das Zentrieren der Punktmenge mittels dem Punkt

$$\bar{X} = \arg \min_{\bar{X} \in \mathbb{R}} \sum_{i=1}^n |X_i - \bar{X}|,$$

4 Approximation mit einfachen Objekten

was dem Median der Datenmenge entspricht.

Beispiel 4.8. Um den Unterschied dieser Verfahren zum Zentrieren der Daten aufzuzeigen, wird auf ein einfaches Beispiel von Keeling und Kunisch [15] zurückgegriffen:

Betrachtet man die Datenmenge $X = \{x_i\}_{1 \leq i \leq n} = \{0, 1, \dots, 1\} \in \mathbb{R}$, sieht man sofort, dass der Wert 0 einen Ausreißer darstellt. Der gewöhnliche Mittelwert würde mittels der Gleichung

$$\bar{X} = \arg \min_{\bar{X} \in \mathbb{R}} \sum_{i=1}^n (x_i - \bar{X})^2 = \frac{n-1}{n}$$

berechnet werden, wobei der Ausreißer beim Berechnen des Medians keinen Effekt zeigen würde, da

$$\bar{X} = \arg \min_{\bar{X} \in \mathbb{R}} \sum_{i=1}^n |x_i - \bar{X}| = 1.$$

Erweitern wir dieses Konzept nun auf eine d -dimensionale Punktmenge, definiert man den geometrischen Median wie folgt:

Definition 4.9. Sei $\{X_i\}_{1 \leq i \leq n} \in \mathbb{R}^d$ eine Punktmenge, dann nennt man einen Punkt X_j aus der gegebenen Menge den geometrischen Median, wenn $\forall i = 1, \dots, n$ gilt, dass

$$\sum_{k=1}^n \|X_k - X_j\|_2 \leq \sum_{k=1}^n \|X_k - X_i\|_2.$$

Sei nun $\{X_i\}_{1 \leq i \leq n}$ eine Punktmenge im \mathbb{R}^d gegeben, dann kann man den geometrischen Median $\bar{X} \in \mathbb{R}^d$ iterativ ermitteln. Für ein $\tau > 0$ führt man folgende Iterationsschritte aus, die eine Matrix $D_l \in \mathbb{R}^{d \times n}$ und mit ihr den Median \bar{X} berechnen:

$$D_{l+1} \vec{e}_j = \frac{D_l \vec{e}_j + \tau(\bar{X}_l - X \vec{e}_j)}{1 + \tau \|\bar{X}_l - X \vec{e}_j\|_2}, \text{ für } j = 1, \dots, n$$

$$\bar{X}_{l+1} = \sum_{j=1}^n \frac{(D_l - \tau X) \vec{e}_j}{1 + \tau \|\bar{X}_l - X \vec{e}_j\|_2} / \sum_{j=1}^n \frac{-\tau}{1 + \tau \|\bar{X}_l - X \vec{e}_j\|_2},$$

wobei X die Matrix mit den Punkten X_i in Spalte i , $(\vec{e}_j)_i = \delta_{ij}$ und somit $X \vec{e}_j$ die j -te Spalte der Matrix X ist. \bar{X}_l und D_l stellen den Median und die Matrix D im l -ten Iterationsschritt dar.

Für die Konvergenzanalyse dieser iterativen Berechnung und die Existenz einer Lösung wird auf [15] verwiesen.

Mit Hilfe des geometrischen Medians können nun Daten, die Ausreißer beinhalten, viel robuster zentriert werden (siehe Abbildung 4.8). Das Ergebnis ist nun besser als in Abbildung 4.5 jedoch noch nicht optimal, weshalb es am Besten mit anderen Varianten wie der der Gewichtsiterationen kombiniert werden sollte.

4 Approximation mit einfachen Objekten

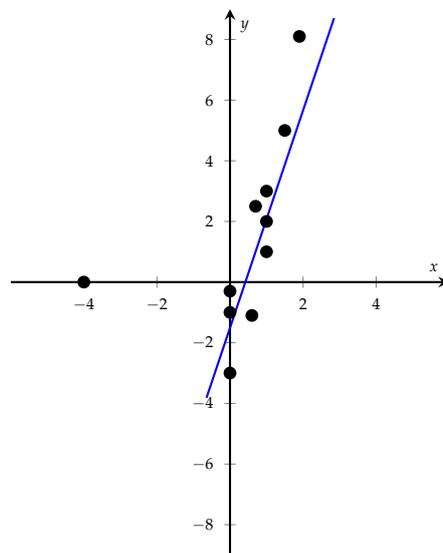


Abbildung 4.8: Approximieren von Datenpunkten mit Ausreißern mittels Zentrierung mit Hilfe des geometrischen Medians

5 Iterative Lösungsverfahren für die Methode der kleinsten Quadrate

Wie schon in Kapitel 3 erwähnt, benötigt die Verwendung der geometrischen Distanz als Fehlerfunktion bei den meisten geometrischen Objekten nichtlineare Berechnungen, für die man meist iterative Verfahren einsetzt.

Betrachtet man eine Punktmenge $\{X_i\}_{1 \leq i \leq n}$, wobei $X_i \in \mathbb{R}^d \forall i$, versucht man für ein geometrisches Objekt O die geometrischen Abstände zu den Punkten zu minimieren. Man versucht Objektparameter \vec{a} zu einem gegebenen Typ eines geometrischen Objektes O zu finden, die die Zielfunktion

$$\sigma(\vec{a})^2 = \sum_{i=1}^n \omega_i d(X_i, O(\vec{a}))^2$$

minimieren, wobei $d(X_i, O(\vec{a}))$ der Abstand vom Punkt X_i zum Objekt O mit Objektparametern \vec{a} und ω_i ein zugehöriges Gewicht ist. Die Lösung eines solchen Minimierungsproblems nennt man auch Methode der kleinsten Quadrate (engl. „least squares fit“).

Viele Methoden zur Lösung eines nichtlinearen Optimierungsproblems sind iterativ. Man versucht, ausgehend von einem Startwert \vec{x}_0 , iterativ die Werte $\vec{x}_1, \vec{x}_2, \dots$ zu berechnen, bis man am Ende gegen ein globales (oder lokales) Minimum konvergiert. Besitzt ein Approximationsproblem mehrere Minima, ist leider nicht immer klar, gegen welches der verwendete Algorithmus konvergiert. Die Konvergenz und Berechnungsdauer hängt besonders vom Startwert \vec{x}_0 ab.

Die meisten Verfahren zur Berechnung dieses Optimierungsproblems haben außerdem Bedingungen eingebaut, die verhindern sollen, dass die Lösung gegen einen Sattelpunkt konvergiert [21].

Auch das Konvergieren gegen lokale Minima ist nicht erwünscht. Chernov und Lesort [8] zeigen jedoch zum Beispiel für das Approximieren mittels eines Kreises, dass die Wahrscheinlichkeit des Auftretens von einem oder mehreren lokalen Minima unter 15% liegt und stetig sinkt, je mehr Punkte man zur Verfügung hat. Wurde die Punktmenge entlang eines Kreises oder Kreisbogens erstellt, dann treten lokale Minima fast gar nicht mehr auf.

5 Iterative Lösungsverfahren für die Methode der kleinsten Quadrate

Neben dem Problem der Konvergenz gegen ein lokales Minimum, treten jedoch noch zwei weitere Probleme bei iterativen Optimierungsalgorithmen auf, die erwähnt werden müssen. Eines dieser Probleme stellt die Divergenz gegen unendlich auf einem steigenden Plateau dar, das zweite, wenn die Iterationsschritte auf einem sehr flachen Tal bzw. Plateau stecken bleiben.

Es gibt viele Methoden zum Lösen eines Problems der kleinsten Quadrate, jede mit gewissen Vor- und Nachteilen. Wir beschäftigen uns hier mit zwei der am meisten verwendeten:

- das Gauss-Newton-Verfahren und
- das Levenberg-Marquardt-Verfahren

Wir betrachten hier die gewichtete Form unseres Approximationsproblems $\sigma^2(\vec{a})$ in Matrixschreibweise.

In den folgenden Abschnitten wird versucht, die Zielfunktion

$$\sigma^2(\vec{a}) = \|P\vec{d}(\vec{a})\|^2 = \vec{d}(\vec{a})^T P^T P \vec{d}(\vec{a}),$$

zu minimieren, wobei

- $\vec{d}(\vec{a}) = (d_1(\vec{a}), \dots, d_n(\vec{a}))$ der Vektor mit allen Distanzen

$$d_i(\vec{a}) = d(X_i, O(\vec{a}))$$

von den gegebenen Punkten $\{X_i\}_{1 \leq i \leq n}$ zum zu approximierenden Objekt O mit den zu ermittelnden Objektparametern \vec{a} sind und

- P eine Gewichtsmatrix darstellt.

Die Matrix P ist eine quadratische Diagonalmatrix

$$P = \begin{pmatrix} \omega_1 & & \\ & \ddots & \\ & & \omega_n \end{pmatrix},$$

wobei ω_i das der Distanz d_i zugehörige Gewicht darstellt.

Bemerkung 5.1. Der Vorteil bei der Verwendung einer Gewichtsmatrix ist, dass man so Ausreißerdaten ein kleines Gewicht geben kann, damit diese das zu approximierende Objekt nicht stark beeinflussen können. Diese Matrix sollte bei jedem Schritt neu erstellt werden und kann die Gewichte zum Beispiel mittels der gemessenen Entfernung des Punktes zum aktuellen Objekt berechnen. Üblicherweise werden die Gewichte zu Beginn alle auf 1 gesetzt, und dann schrittweise angepasst (siehe Abschnitt 4.2.3).

Um ein Minimum zu finden, betrachten wir die erste Ableitung der Zielfunktion $\sigma(\vec{a})^2$

$$\frac{\partial \sigma(\vec{a})^2}{\partial \vec{a}},$$

5 Iterative Lösungsverfahren für die Methode der kleinsten Quadrate

mit den Objektparametern $\vec{a} = \begin{pmatrix} a_1 \\ \vdots \\ a_p \end{pmatrix}$, und setzen sie gleich 0. Das Hauptkriterium, das für das Auffinden eines Minimums gelten muss, ist somit gegeben durch

$$\left(\frac{\partial \sigma(\vec{a})}{\partial \vec{a}}\right)^T = \left(\frac{\partial (\|P\vec{d}(\vec{a})\|^2)}{\partial \vec{a}}\right)^T = 2J(\vec{a})^T P^T P \vec{d}(\vec{a}) \stackrel{!}{=} (\vec{0}),$$

mit dem Distanzvektor $\vec{d}(\vec{a}) = \begin{pmatrix} d_1(\vec{a}) \\ \vdots \\ d_n(\vec{a}) \end{pmatrix}$, den Distanzfunktionen $d_i(\vec{a}) = d(X_i, O(\vec{a}))$ von den gegebenen Punkten $\{X_i\}_{1 \leq i \leq n}$ zum zu approximierenden Objekt O und der Jakobi-Matrix

$$J(\vec{a}) = \begin{pmatrix} \frac{\partial d_1}{\partial a_1} & \cdots & \frac{\partial d_1}{\partial a_p} \\ \vdots & & \vdots \\ \frac{\partial d_n}{\partial a_1} & \cdots & \frac{\partial d_n}{\partial a_p} \end{pmatrix}.$$

5.1 Die Newton-Methode

Der Vollständigkeit halber wird hier noch kurz auf die Newton-Methode (auch bekannt als Newton-Raphson-Methode) eingegangen, da die weiteren zwei Verfahren, die für unseren Fall eines Minimierungsproblems betrachtet werden, von dieser abgeleitet sind.

Beginnen wir hier mit dem univariaten Fall und erweitern diesen dann auf den multivariaten Fall. Der univariate Fall der Newton-Raphson-Methode versucht eine Gleichung $f(x) = 0$ mittels einer Anfangslösung x_0 zu lösen. In Abbildung 5.1 sieht man, wie die Newton-Methode die Lösung für die obere Gleichung mittels Tangenten sucht.

Man betrachtet also die Tangentengleichung für die Tangente t im Punkt x_0

$$f'(x_0) = \frac{t(x_1) - f(x_0)}{x_1 - x_0}$$

Da hier versucht wird ein Ergebnis der Art $f(x_1) \approx t(x_1) = 0$ zu bekommen, kann man die Gleichung umformen zu

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}.$$

Somit ist der Iterationsschritt der Newton-Methode gegeben durch

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \quad \forall k \geq 0.$$

5 Iterative Lösungsverfahren für die Methode der kleinsten Quadrate

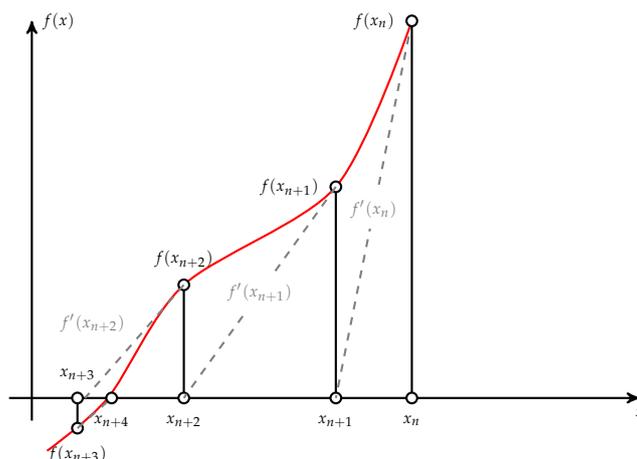


Abbildung 5.1: Graphische Darstellung der univariaten Newton-Methode

Nun zur Konvergenz dieses Verfahrens:

Satz 5.2. Sei $f : [a, b] \rightarrow \mathbb{R}$ eine zweimal differenzierbare konvexe Funktion mit $f(a) < 0$ und $f(b) > 0$. Dann gilt für ein beliebiges $x_0 \in [a, b]$ mit $f(x_0) \geq 0$ und $n \in \mathbb{N}$, dass die Folge

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

wohldefiniert ist und monoton fallend gegen $\zeta \in (a, b)$ mit $f(\zeta) = 0$ konvergiert.

Beweis. Die Existenz und Eindeutigkeit von ζ mit $f(\zeta) = 0$ ist durch den Mittelwertsatz und der Eigenschaft, dass f konvex ist, gegeben. Sei nun $x_0 \in [a, b]$ mit $f(x_0) \geq 0$. Nach den Voraussetzungen für f gilt $x_0 \geq \zeta$.

Durch Induktion wird im Folgenden bewiesen, dass für eine Folge (x_n) mit

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

gilt, dass $f(x_{n+1}) \geq 0$ und die Folge monoton fallend ist.

Induktionsanfang: Für $n = 0$ gilt das Gewünschte.

Induktionsschritt: $n \rightarrow n + 1$:

Da $x_n \geq \zeta$ gilt, folgt $f'(x_n) \geq f'(\zeta) > 0$, woraus sich

$$\frac{f(x_n)}{f'(x_n)} \geq 0$$

ergibt.

5 Iterative Lösungsverfahren für die Methode der kleinsten Quadrate

Mit $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$ ergibt sich schließlich

$$x_{n+1} \leq x_n.$$

Da f konvex ist, ist f' monoton wachsend. Mit der Hilfsfunktion

$$g(x) = f(x) - f(x_n) - (x - x_n)f'(x_n),$$

folgt wegen

$$g'(x) = f'(x) - f'(x_n) \leq 0$$

für $x \leq x_n$ und $g(x_n) = 0$, dass $g(x) \geq 0$ für $x \leq x_n$ ist.

Insbesondere gilt

$$0 \leq g(x_{n+1}) = f(x_{n+1}) - f(x_n) - (x_{n+1} - x_n)f'(x_n) = f(x_{n+1}),$$

womit gezeigt ist, dass die Folge (x_n) monoton fällt und nach unten durch ζ beschränkt ist. Daraus folgt die Existenz des Grenzwerts $\lim_{n \rightarrow \infty} x_n = x^*$.

Mittels Fixpunktiteration bekommt man schließlich $f(x^*) = 0$ und auf Grund der Eindeutigkeit der Nullstelle folgt $x^* = \zeta$. \square

Bemerkung 5.3. Der Beweis für f konkav oder $f(a) > 0$ und $f(b) < 0$ wird analog geführt.

Erweitern wir dieses Verfahren nun auf den multivariaten Fall: Hier wird das soeben beschriebene Verfahren auf eine differenzierbare Funktion

$$f(x_1, \dots, x_n) : \mathbb{R}^n \rightarrow \mathbb{R}^n$$

mit $f = (f_1, \dots, f_n)$ angewendet, wodurch sich der Iterationsschritt ändert zu

$$\vec{x}_{k+1} = \vec{x}_k - J(\vec{x}_k)^{-1} f(\vec{x}_k) \quad \forall k \geq 0,$$

mit der Jakobimatrix

$$J(\vec{x}_k) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{pmatrix}.$$

Dieser multivariate Fall wird in den nächsten zwei Verfahren leicht abgewandelt, um es auch auf unseren Fall der Minimierung anwenden zu können.

5.2 Das Gauss-Newton-Verfahren

Die erste Methode, die hier betrachtet wird, um nichtlineare Approximationsprobleme effizient zu lösen, ist das Gauss-Newton-Verfahren.

Versuchen wir nun zuerst das Gauss-Verfahren auf unseren Fall der Minimierung zu erweitern:

Man setze die Funktion $f(x)$, für die die Newton-Methode die Nullstellen suchen sollte, auf die erste Ableitung der zuvor definierten Zielfunktion $\sigma^2(\vec{a}) = \|P\vec{d}(\vec{a})\|^2$ mit Objektparametern \vec{a} , dem Distanzvektor $\vec{d}(\vec{a}) = (d_1(\vec{a}) \dots d_n(\vec{a}))$, der Distanz

$$d_i(\vec{a}) = d(X_i, O(\vec{a}))$$

jedes gegebenen Punktes zum zu approximierenden Typ eines geometrischen Objektes O mit Objektparametern \vec{a} und einer wie zuvor definierten Gewichtsmatrix P .

Dies ergibt mit den Objektparametern \vec{a}_k aus dem k -ten Iterationsschritt

$$f(\vec{a}_k) = \left(\frac{\partial \sigma^2(\vec{a}_k)}{\partial \vec{a}_k} \right),$$

womit sich der Iterationsschritt der Newton-Methode ersetzen lässt durch

$$a_{k+1} = a_k - \alpha \left(\frac{\partial^2 \sigma(\vec{a}_k)^2}{\partial \vec{a}_k^2} \right)^{-1} \left(\frac{\partial \sigma(\vec{a}_k)^2}{\partial \vec{a}_k} \right)^T$$

mit einer Schrittweite α .

Dies zeigt, dass man für das Gauss-Verfahren die zweite Ableitung der Zielfunktion benötigt. Um dies zu vermeiden ersetzt man bei der Gauss-Newton-Methode den Distanzvektor $\vec{d}(\vec{a}_{k+1})$ mit Objektparametern \vec{a}_{k+1} aus dem $k+1$ -ten Iterationsschritt mit einer Linearisierung der Form

$$\vec{d}(\vec{a}_{k+1}) \approx \vec{d}(\vec{a}_k) + J(\vec{a}_k)(\vec{a}_{k+1} - \vec{a}_k).$$

Mit dieser Linearisierung erhält man die approximierte Zielfunktion

$$\sigma(\vec{a}_{k+1})^2 = \|P\vec{d}(\vec{a}_{k+1})\|^2 \approx \|P(\vec{d}(\vec{a}_k) + J(\vec{a}_k)(\vec{a}_{k+1} - \vec{a}_k))\|^2 = \sigma(\vec{a}_{k+1})_{\approx}^2.$$

Das Minimum wird wieder erreicht, wenn man die erste Ableitung der approximierten Zielfunktion gleich Null setzt. Betrachtet man also die Ableitung von $\sigma(\vec{a})_{\approx}^2$ nach \vec{a} und setzt sie gleich Null, erhält man

$$\left(\frac{\partial \sigma(\vec{a})_{\approx}^2}{\partial \vec{a}} \right) = 2J(\vec{a})^T P^T P \vec{d}(\vec{a}) + 2J(\vec{a})^T P^T P J(\vec{a})(\vec{a}_k - \vec{a}) = 0.$$

5 Iterative Lösungsverfahren für die Methode der kleinsten Quadrate

Durch eine einfache Umformung ergibt sich der iterative Schritt des Gauss-Newton-Verfahrens für die neuen Objektparameter \vec{a}_{k+1}

$$(J(\vec{a}_k)^T P^T P J(\vec{a}_k))(\vec{a}_{k+1} - \vec{a}_k) = -J(\vec{a}_k)^T P^T P \vec{d}(\vec{a}_k). \quad (5.1)$$

Das Minimum wird somit angenommen für

$$\vec{a}_{k+1} = \vec{a}_k - \left(J(\vec{a}_k)^T P^T P J(\vec{a}_k) \right)^{-1} \left(J(\vec{a}_k)^T P^T P \vec{d}(\vec{a}_k) \right),$$

womit man die Iterationsformel des Gauss-Newton-Verfahrens bekommt. [13]

Diese Iterationen werden so lange ausgeführt, bis für ein $\varepsilon > 0$ eine der folgenden Abbruchkriterien erfüllt ist:

- $\|J(\vec{a}_k)^T P^T P \vec{d}(\vec{a}_k)\| < \varepsilon$,
- $\|\vec{a}_{k+1} - \vec{a}_k\| < \varepsilon$, oder
- $\sigma(\vec{a}_k)^2 - \sigma(\vec{a}_{k+1})^2 < \varepsilon$.

Das erste Kriterium beschreibt den Fall, wenn die erste Ableitung der Zielfunktion einen Wert kleiner ε erreicht und somit nahe an einem Minimum liegt. Das zweite Stoppkriterium tritt in Kraft, wenn sich die Objektparameter nur mehr minimal ändern, und das dritte Kriterium wird für den Abbruch des Algorithmus verwendet, wenn sich der Fehler nur mehr sehr langsam bis gar nicht mehr verkleinert.

Der Algorithmus für das Gauss-Newton-Verfahren lautet somit wie in Abbildung 5.2 ersichtlich.

Das Gauss-Newton-Verfahren kann jedoch unter Umständen Probleme mit der Konvergenz haben. Dies kann besonders dann auftreten, wenn die Startlösung nicht nahe am Minimum liegt.

5.3 Das Levenberg-Marquardt-Verfahren

Eine besonders in der Messtechnik verwendete Methode zum Lösen eines Approximationsproblems ist das Levenberg-Marquardt-Verfahren. Diese Methode wird auch für die Lösung des in Kapitel 6 beschriebenen Approximationsproblems verwendet.

Da die zuvor beschriebene Gauss-Newton-Methode teilweise Probleme mit der Konvergenz gegen ein Minimum hat, schlug Levenberg [18] und später Marquardt [22] eine Korrektur des Berechnungsschrittes der Form

$$(J(\vec{a}_k)^T P^T P J(\vec{a}_k) + \mu I)(\vec{a}_{k+1} - \vec{a}_k) = -J(\vec{a}_k)^T P^T P \vec{d}(\vec{a}_k) \quad (5.2)$$

vor, wobei der Korrekturfaktor $\mu \geq 0$ folgende Effekte hervorbringen sollte:

5 Iterative Lösungsverfahren für die Methode der kleinsten Quadrate

Eingabe: eine Distanzfunktion $\vec{d}(\vec{a})$, initiale Modellparameter a_0 , Abbruchparameter ε_1 und ε_2 , Anzahl der maximalen Iterationen k_{\max}

Ausgabe: Die optimierten Modellparameter \vec{a}

Initialisierung

$k := 0$

$\vec{a} := \vec{a}_0$

$A := J(\vec{a})^T J(\vec{a})$

$\vec{g} := J(\vec{a})^T \vec{d}(\vec{a})$

found := ($\|\vec{g}\|_\infty \leq \varepsilon_1$)

while $!(found)$ AND $k < k_{\max}$ **do**

$k := k + 1$

 Löse $A(\vec{a} - \vec{a}_k) = -\vec{g}$

if $\|(\vec{a} - \vec{a}_k)\| \leq \varepsilon_2$ **then**

 found := true

end

else

$\vec{a}_{\text{new}} := \vec{a} + (\vec{a} - \vec{a}_k)$

$\vec{a} := \vec{a}_{\text{new}}$

$A := J(\vec{a})^T J(\vec{a})$

$\vec{g} := J(\vec{a})^T \vec{d}(\vec{a})$

 found := ($\|\vec{g}\|_\infty \leq \varepsilon_1$)

end

end

Abbildung 5.2: Gauss-Newton-Algorithmus

- Für alle $\mu > 0$ ist die Koeffizientenmatrix positiv definit, wodurch abgesichert wird, dass der Iterationsschritt eine absteigende Richtung findet.
- Für große Werte von μ erhält man

$$(\vec{a}_{k+1} - \vec{a}_k) \approx -\frac{1}{\mu} J(\vec{a}_k)^T P^T P \vec{d}(\vec{a}_k),$$

was einen großen Schritt in die am stärksten abfallende Richtung darstellt. Dies ist gut, wenn man noch weit vom Minimum entfernt ist.

- Für sehr kleine μ gilt

$$(\vec{a}_{k+1} - \vec{a}_k) \approx (\vec{a}^* - \vec{a}_k),$$

mit \vec{a}^* die optimale Lösung der Fehlerfunktion. Dies ist vor allem in den letzten Iterationsschritten wichtig, wenn man das Minimum schon fast erreicht hat.

Der Korrekturfaktor beeinflusst also sowohl die Richtung als auch die Schrittgröße für jeden Iterationsschritt. [21]

5 Iterative Lösungsverfahren für die Methode der kleinsten Quadrate

Der Startwert für μ sollte abhängig von den Einträgen der Matrix $A = (a_{ii}) = J(\vec{a}_0)^T J(\vec{a}_0)$ gewählt werden, zum Beispiel

$$\mu_0 = \tau \cdot \max_i a_{ii},$$

wobei der Parameter τ frei gewählt werden kann.

Der komplette Algorithmus kann nun also wie in Abbildung 5.3 aufgestellt werden (hier der Anschaulichkeit halber ohne Gewichtsmatrix P , die jedoch leicht eingebunden werden kann) [21].

Der Parameter ρ im Algorithmus ist das so genannte Nutzen-Verhältnis (engl. „gain ratio“), wobei

$$L(\vec{a} - \vec{a}_{\text{new}}) = \vec{d}(\vec{a}) + 2(\vec{a} - \vec{a}_{\text{new}})^T J(\vec{a})^T \vec{d}(\vec{a}) + (\vec{a} - \vec{a}_{\text{new}})^T J(\vec{a})^T J(\vec{a})(\vec{a} - \vec{a}_{\text{new}})$$

eine quadratische Approximation der Zielfunktion (analog zum Gauss-Newton-Verfahren) im Punkt \vec{a} darstellt.

Somit bekommt man zusammenfassend für den Nenner im Nutzen-Verhältnis:

$$\begin{aligned} L(\vec{0}) - L(\vec{a} - \vec{a}_{\text{new}}) &= -2(\vec{a} - \vec{a}_{\text{new}})^T J(\vec{a})^T \vec{d}(\vec{a}) - (\vec{a} - \vec{a}_{\text{new}})^T J(\vec{a})^T J(\vec{a})(\vec{a} - \vec{a}_{\text{new}}) \\ &= -(\vec{a} - \vec{a}_{\text{new}})^T (2J(\vec{a})^T \vec{d}(\vec{a}) + (J(\vec{a})^T J(\vec{a}) + \mu I - \mu I)(\vec{a} - \vec{a}_{\text{new}})) \\ &= (\vec{a} - \vec{a}_{\text{new}})^T (\mu I(\vec{a} - \vec{a}_{\text{new}}) - J(\vec{a})^T \vec{d}(\vec{a})), \end{aligned}$$

da ohne Gewichtsmatrix P für die Gleichung (5.1) gilt, dass

$$(J(\vec{a}_k)^T J(\vec{a}_k) + \mu I)(\vec{a} - \vec{a}_k) = -J(\vec{a}_k)^T \vec{d}(\vec{a}_k).$$

Ein großer Wert für ρ weist auf einen guten Iterationsschritt hin, so dass der Parameter μ verkleinert werden kann, wobei μ jedoch bei einem kleinen Wert für ρ vergrößert werden sollte, um schneller im Richtung des gesuchten Minimums zu gelangen.

Der Parameter k_{max} wird als zusätzliches Abbruchkriterium eingeführt, um Endlosschleifen zu verhindern. Dieser kann beliebig groß gewählt werden.

Nun zur Konvergenz des Verfahrens:

Angenommen im k -ten Iterationsschritt gilt für die Fehlerfunktion, dass sie einen größeren Wert liefert, als im Schritt zuvor, das heißt

$$\sigma_{k+1}^2 > \sigma_k^2,$$

was unter Umständen zur Divergenz des Verfahrens führen könnte.

Hier bewies jedoch Marquardt folgenden Satz, mit dem die Konvergenz des Verfahrens gesichert ist [22]:

5 Iterative Lösungsverfahren für die Methode der kleinsten Quadrate

Eingabe: eine Distanzfunktion $\vec{d}(\vec{a})$, initiale Modellparameter a_0 , Faktor τ , Abbruchparameter ε_1 und ε_2 , Anzahl der maximalen Iterationen k_{\max}

Ausgabe: Die optimierten Modellparameter \vec{a}

Initialisierung $k := 0; \nu := 2; \vec{a} := \vec{a}_0$

$A := J(\vec{a})^T J(\vec{a})$

$\vec{g} := J(\vec{a})^T \vec{d}(\vec{a})$

found := ($\|\vec{g}\|_\infty \leq \varepsilon_1$)

$\mu := \tau \cdot \max a_{ii}$

while $!(found)$ AND $k < k_{\max}$ **do**

$k := k + 1$

 Löse $(A + \mu I)(\vec{a} - \vec{a}_k) = -\vec{g}$

if $\|(\vec{a} - \vec{a}_k)\| \leq \varepsilon_2(\|\vec{a}\| + \varepsilon_2)$ **then**

 found := true

end

else

$\vec{a}_{\text{new}} := \vec{a} + (\vec{a} - \vec{a}_k)$

$\rho := \frac{\vec{d}(\vec{a}) - \vec{d}(\vec{a}_{\text{new}})}{L(0) - L(\vec{a} - \vec{a}_{\text{new}})}$

if $\rho > 0$ **then**

$\vec{a} := \vec{a}_{\text{new}}$

$A := J(\vec{a})^T J(\vec{a})$

$\vec{g} := J(\vec{a})^T \vec{d}(\vec{a})$

 found := ($\|\vec{g}\|_\infty \leq \varepsilon_1$)

$\mu := \mu \cdot \max \left\{ \frac{1}{3}, 1 - (2\rho - 1)^3 \right\}$

$\nu := 2$

end

else

$\mu := \mu \cdot \nu$

$\nu := 2\nu$

end

end

end

Abbildung 5.3: Levenberg-Marquardt-Algorithmus

Satz 5.4. Für den Berechnungsschritt

$$(A + \mu I)(\vec{a}_{k+1} - \vec{a}_k) = \vec{b}$$

des Levenberg-Marquardt-Verfahrens ist es immer möglich, den Parameter μ so zu wählen, dass für die quadratische Fehlerfunktion σ^2 gilt, dass

$$\sigma_{k+1}^2 \leq \sigma_k^2.$$

5.4 Lösungsmethoden für lineare Gleichungssysteme

Wie die oben beschriebenen Verfahren gezeigt haben, lässt sich das gegebene Minimierungsproblem lösen, indem man Gleichungen der Form

$$A^T A \vec{x} = A^T \vec{b}$$

löst. Dies beschreibt im Gauss-Newton-Verfahren (siehe Abschnitt 5.2) den Berechnungsschritt 5.1 sowie im Levenberg-Marquardt-Verfahren den erweiterten Berechnungsschritt 5.2.

Um lineare Gleichungssysteme zu lösen, gibt es mehrere Methoden, zu denen folgende drei Zerlegungen gehören:

- Die Cholesky-Zerlegung,
- die QR-Zerlegung und
- die Singulärwertzerlegung.

5.4.1 Die Cholesky-Zerlegung

Die erste hier betrachtete Methode zur Lösung von linearen Gleichungssystemen ist die Cholesky-Zerlegung. Für diese Zerlegung benötigt man die Eigenschaft, dass die zu zerlegende Matrix A symmetrisch und positiv definit ist:

Definition 5.5. Eine symmetrische Matrix $A \in \mathbb{R}^{n \times n}$ (das heißt also $A^T = A$) nennt man positiv definit, wenn $\forall \vec{x} \in \mathbb{R}^n \setminus \{\vec{0}\}$ gilt, dass

$$\vec{x}^T A \vec{x} > 0.$$

Definition 5.6. Eine Hauptuntermatrix $B \in \mathbb{R}^{k \times k}$ einer Matrix $A \in \mathbb{R}^{n \times n}$, mit $k \leq n$, geht aus dem Streichen der i -ten Zeile und i -ten Spalte von A , $\forall i \in J \subset \{1, \dots, n\}$, hervor.

Bemerkung 5.7. Eine symmetrische, positiv definite Matrix A besitzt folgende Eigenschaften:

- A ist nicht-singulär, das heißt A besitzt eine Inverse A^{-1} .

5 Iterative Lösungsverfahren für die Methode der kleinsten Quadrate

- Sei $B \in \mathbb{R}^{n \times m}$ eine Matrix mit vollem Rang und $n > m$, dann ist $B^T A B$ ebenfalls symmetrisch und positiv definit.
- Alle Hauptuntermatrizen von A sind symmetrisch und positiv definit.
- A besitzt positive reelle Eigenwerte und eine Orthonormalbasis aus Eigenvektoren.

Satz 5.8. Sei $A \in \mathbb{R}^{n \times n}$ symmetrisch und positiv definit, dann existiert eine Zerlegung

$$A = B B^T,$$

mit einer unteren Dreiecksmatrix $B = (b_{i,j})_{1 \leq i,j \leq n}$ mit positiven Diagonaleinträgen $b_{i,i} \forall i$. Diese Zerlegung wird auch Cholesky-Zerlegung genannt.

Beweis. Induktion nach n :

$n = 1$: trivial

$n - 1 \rightarrow n$: Sei nun $A \in \mathbb{R}^{n \times n}$ mit

$$A = \begin{pmatrix} \tilde{A} & \vec{x} \\ \vec{x}^T & a \end{pmatrix},$$

wobei $\tilde{A} \in \mathbb{R}^{(n-1) \times (n-1)}$. Nachdem jede Hauptuntermatrix einer symmetrisch, positiv definiten Matrix wieder symmetrisch und positiv definit ist, besitzt \tilde{A} nach Induktionsvoraussetzung eine Cholesky-Zerlegung

$$\tilde{A} = \tilde{B} \tilde{B}^T.$$

Betrachte das Matrixprodukt

$$B_1^{-1} A B_1^{-T} = \begin{pmatrix} \tilde{B}^{-1} & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \tilde{A} & \vec{x} \\ \vec{x}^T & a \end{pmatrix} \begin{pmatrix} \tilde{B}^{-T} & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} I & \vec{b} \\ \vec{b}^T & a \end{pmatrix},$$

mit $\vec{b} = \tilde{B}^{-1} \vec{x}$.

Im nächsten Schritt versucht man, \vec{b} mit Hilfe einer unteren Dreiecksmatrix zu eliminieren:

$$B_2^{-1} B_1^{-1} A B_1^{-T} B_2^{-T} = \begin{pmatrix} I & 0 \\ -\vec{b}^T & 1 \end{pmatrix} \begin{pmatrix} I & \vec{b} \\ \vec{b}^T & a \end{pmatrix} \begin{pmatrix} I & -\vec{b} \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} I & 0 \\ 0 & a - \vec{b}^T \vec{b} \end{pmatrix} = \begin{pmatrix} I & 0 \\ 0 & a - \vec{x}^T A^{-1} \vec{x} \end{pmatrix}.$$

Diese Matrix ist nun wieder positiv definit und der Term $a - \vec{x}^T A^{-1} \vec{x} = \gamma^2$ ist positiv mit $\gamma \in \mathbb{R}$. Mit $\gamma > 0$ und der Matrix

$$B_3 = \begin{pmatrix} I & 0 \\ 0 & \gamma \end{pmatrix}$$

gilt nun

$$B_2^{-1} B_1^{-1} A B_1^{-T} B_2^{-T} = B_3 B_3^T$$

5 Iterative Lösungsverfahren für die Methode der kleinsten Quadrate

und man bekommt die gewünschte Cholesky-Zerlegung $A = BB^T$ mit

$$B = B_1 B_2 B_3 = \begin{pmatrix} \tilde{B} & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} I & 0 \\ \vec{b}^T & 1 \end{pmatrix} \begin{pmatrix} I & 0 \\ 0 & \gamma \end{pmatrix} = A = \begin{pmatrix} \tilde{B} & 0 \\ \vec{x}^T \tilde{B}^{-1} & \gamma \end{pmatrix}.$$

□

Betrachten wir nun die Matrix A der gegebenen Gleichung

$$A^T A \vec{x} = A^T \vec{b},$$

folgt, dass wenn A vollen Rang hat, das Matrixprodukt $A^T A$ der Gleichung quadratisch und positiv definit ist.

Nun lässt sich die Cholesky-Zerlegung wie folgt anwenden:

- Berechnen der Matrix $A^T A$ und des Vektors $A^T \vec{b}$
- Berechnen der Cholesky-Zerlegung

$$A^T A = R^T R$$

- Lösen des Systems mit der unteren Dreiecksmatrix

$$R^T \vec{y} = A^T \vec{b}$$

- Lösen des Systems mit der oberen Dreiecksmatrix

$$R \vec{x} = \vec{y}$$

5.4.2 Die QR-Zerlegung

Wir betrachten eine weitere Zerlegungsmethode für Matrizen, nämlich die QR-Zerlegung.

Satz 5.9. Sei A eine $n \times m$ Matrix. Dann besitzt A eine Zerlegung der Form

$$A = QR,$$

mit einer $n \times n$ Orthogonalmatrix Q , das heißt also $Q^T Q = I$, und einer oberen $n \times m$ Dreiecksmatrix R .

Dieser Satz wird mittels Konstruktion über das Gram-Schmidt Verfahren bewiesen.

Durch Einsetzen der Zerlegung $A = QR$ ist die Gleichung

$$A^T A \vec{x} = A^T \vec{b}$$

äquivalent zu

$$R^T Q^T Q R \vec{x} = R^T Q^T \vec{b}.$$

5 Iterative Lösungsverfahren für die Methode der kleinsten Quadrate

Hat die Matrix A vollen Rang, ist R invertierbar und man erhält die Gleichung

$$R\vec{x} = Q^T\vec{b}.$$

Diese Zerlegung ist also ebenfalls ausgelegt auf Matrizen A mit vollem Rang.

Der Lösungsvorgang für das lineare Gleichungssystem $A\vec{x} = \vec{b}$ lässt sich somit zusammenfassen zu:

- Berechnen der QR-Zerlegung $A = QR$
- Berechnen des Vektors $Q^T\vec{b}$
- Lösen des Systems mit der oberen Dreiecksmatrix

$$R\vec{x} = Q^T\vec{b}$$

5.4.3 Die Singulärwertzerlegung

Ist die Matrix $A \in \mathbb{R}^{n \times m}$ nicht quadratisch oder ist sie quadratisch und besitzt keinen vollen Rang, das heißt $rk(A) < n$, würde die Cholesky-Zerlegung fehlschlagen. Da in so einem Fall die Lösung für ein Problem der kleinsten Quadrate möglicherweise keine eindeutige Lösung besitzt, kann mit Hilfe der Singulärwertzerlegung unter den möglichen Lösungen diejenige ausgewählt werden, die die kleinste Norm besitzt („Lösung im Sinne der minimalen Norm“, engl. „Minimum Norm Solution“).

Hierfür betrachtet man die Singulärwertzerlegung der Inversen A^+ der Matrix A :

$$A^+ = V\Sigma^+U^T,$$

mit $\Sigma^+ = \text{diag}\{\sigma_1^{-1}, \dots, \sigma_n^{-1}\}$ eine Matrix, die aus den Singulärwerten $\sigma_1, \dots, \sigma_n$ gebildet wird, V eine Matrix mit den linken Singulärvektoren und U eine Matrix mit den rechten Singulärvektoren von A^+ .

Im Fall, dass die Matrix A keinen vollen Rang besitzt oder nicht quadratisch ist betrachtet man die Pseudoinverse A^+ von A . Sei hierfür $A = U\Sigma V^T$ die Singulärwertzerlegung von A mit $rk(A) = r < n$. Dann besitzt die Diagonalmatrix Σ die Form

$$\Sigma = \begin{pmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{pmatrix},$$

wobei $\Sigma_1 = \text{diag}\{\sigma_1, \dots, \sigma_r\}$ mit $\sigma_1 \geq \dots \geq \sigma_r > 0$ und den Matrizen $U = (U_1 \ U_2)$ und $V = (V_1 \ V_2)$ dementsprechend aufgeteilt. Daraus ergibt sich die Pseudoinverse

$$A^+ = V\Sigma^+U^T = V_1\Sigma_1^{-1}U_1^T,$$

mit

$$\Sigma^+ = \begin{pmatrix} \Sigma_1^{-1} & 0 \\ 0 & 0 \end{pmatrix}.$$

5 Iterative Lösungsverfahren für die Methode der kleinsten Quadrate

Die Pseudoinverse existiert immer, da Σ_1 nicht singulär ist. Mit Hilfe der Zerlegung und der Orthogonalität von U , lässt sich das Problem der kleinsten Quadrate darstellen als

$$\begin{aligned}\|\vec{b} - A\vec{x}\|_2^2 &= \|UU^T\vec{b} - U\Sigma V^T\vec{x}\|_2^2 \\ &= \|UU^T\vec{b}\|_2^2 - 2\langle UU^T\vec{b}, U\Sigma V^T\vec{x}\rangle + \|U\Sigma V^T\vec{x}\|_2^2 \\ &= \|U^T\vec{b}\|_2^2 - 2\langle U^T\vec{b}, \Sigma V^T\vec{x}\rangle + \|\Sigma V^T\vec{x}\|_2^2 \\ &= \|U^T\vec{b} - \Sigma V^T\vec{x}\|_2^2 \\ &= \left\| \begin{pmatrix} U_1^T\vec{b} - \Sigma_1\vec{y}_1 \\ U_2^T\vec{b} \end{pmatrix} \right\|_2^2 \\ &= \|U_1^T\vec{b} - \Sigma_1\vec{y}_1\|_2^2 + \|U_2^T\vec{b}\|_2^2,\end{aligned}$$

mit $\vec{y} = V^T\vec{x} = \begin{pmatrix} V_1^T \\ V_2^T \end{pmatrix} \vec{x} = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$.

Dadurch folgt, dass

$$\|\vec{b} - A\vec{x}\|_2 \geq \|U_2^T\vec{b}\|_2 \quad \forall \vec{x} \in \mathbb{R}^n.$$

Gleichheit wird erreicht, wenn

$$\vec{x} = V\vec{y} = (V_1 \ V_2) \begin{pmatrix} \Sigma_1^{-1}U_1^T\vec{b} \\ \vec{y}_2 \end{pmatrix} = A^+\vec{b} + V_2\vec{y}_2,$$

für ein $\vec{y}_2 \in \mathbb{R}^{n-r}$. Diese Lösung ist eindeutig, wenn A vollen Rang besitzt ($r = n$). Ansonsten sei

$$\vec{z} = V\vec{y} = (V_1 \ V_2) \begin{pmatrix} \Sigma_1^{-1}U_1^T\vec{b} \\ \vec{y}_2 \end{pmatrix} = A^+\vec{b} + V_2\vec{y}_2,$$

für ein \vec{y}_2 nicht Null und $\vec{x} = A^+\vec{b}$. Dann ist

$$\|\vec{z}\|_2^2 = \|A^+\vec{b}\|_2^2 + \|\vec{y}_2\|_2^2 > \|\vec{x}\|_2^2,$$

so dass

$$\vec{x} = A^+\vec{b} = V_1\Sigma_1^{-1}U_1^T\vec{b}$$

die Lösung im Sinne der minimalen Norm ist. [17]

Somit lässt sich die Berechnung des linearen Gleichungssystems $A\vec{x} = \vec{b}$ mittels der Singulärwertzerlegung zusammenfassen als:

- Berechnen der Singulärwertzerlegung $A = U\Sigma V^T$
- Berechnen des Vektors $U_1^T\vec{b}$
- Lösen des Gleichungssystems mit der Diagonalmatrix

$$\Sigma_1\vec{y} = U_1^T\vec{b}$$

- Lösen des Gleichungssystems

$$\vec{x} = V_1\vec{y}$$

5.4.4 Vergleich der drei Zerlegungsmethoden

Ein genauer Vergleich dieser drei Methoden ist bei vielen Autoren nachzulesen (siehe zum Beispiel [19], [9], [20] oder [4]), weshalb hier nur ein kleiner Überblick gegeben wird.

Die Cholesky-Zerlegung ist die schnellste der drei Methoden, jedoch die am wenigsten stabile. Sie funktioniert nur in dem Fall gleich gut wie die anderen zwei, wenn die Matrix gut konditioniert ist.

Sollte die Matrix nicht gut konditioniert sein, jedoch vollen Rang besitzen, wird die QR-Zerlegung als beste Möglichkeit angesehen. Diese ist auch das am meisten verwendete Verfahren zur Bestimmung von linearen Problemen der kleinsten Quadrate.

Die Singulärwertzerlegung ist die aufwendigste der drei Methoden, jedoch die einzige, die mit Matrizen umgehen kann, die keinen vollen Rang besitzen.

5.5 Approximation einer Punktmenge mit einem Kreis im \mathbb{R}^2

Bevor wir im nächsten Kapitel auf ein praktisches Anwendungsbeispiel der Methode der kleinsten Quadrate eingehen, betrachten wir zuerst eine Implementation einer Kreisapproximation von Ahn [2].

Es wird versucht, die Summe der quadratischen Abstände von gegebenen Punkten $\{X_i\}_{1 \leq i \leq n}$ zu einem Kreis O zu minimieren, was die Zielfunktion

$$\sum_{i=1}^n d(X_i, O)^2$$

ergibt, wobei $d(X_i, O)$ den Abstand des Punktes X_i zum Kreis O darstellt.

Ein Kreis mit Radius r und Mittelpunkt X_0 im \mathbb{R}^2 kann in impliziter Form dargestellt werden als

$$F(\vec{a}, X) \triangleq \|X - X_0\|^2 - r^2 = 0,$$

mit $\vec{a}^T = (r, X_0^T)$.

Ein Punkt X'_i am Kreis mit der minimalen Distanz zu einem Punkt X_i wird berechnet durch

$$X'_i = X_0 + r \frac{X_i - X_0}{\|X_i - X_0\|},$$

und die Distanz zwischen diesen Punkten ist gegeben als

$$d_i = \|X_i - X_0\| - r.$$

5 Iterative Lösungsverfahren für die Methode der kleinsten Quadrate

Somit ist es relativ einfach möglich, die Abstände zu den gegebenen Punkten $\{X_i\}_{1 \leq i \leq n}$ zu berechnen.

Als Nächstes benötigt man die Jakobimatrix $J_{d_i, \vec{a}}$ der Distanz d_i . Diese ist gegeben durch

$$\begin{aligned} J_{d_i, \vec{a}} &= \frac{\partial d_i}{\partial \vec{a}} \\ &= \frac{\partial}{\partial \vec{a}} (\|X_i - X_0\| - r) \\ &= -\frac{(X_i - X_0)^T}{\|X_i - X_0\|} \frac{\partial X_0}{\partial \vec{a}} - \frac{\partial r}{\partial \vec{a}} \end{aligned}$$

wobei

$$\frac{\partial r}{\partial \vec{a}} = (1 \quad 0 \quad 0)$$

und

$$\frac{\partial X_0}{\partial \vec{a}} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Nimmt man nun den Berechnungsschritt des Gauss-Newton-Verfahrens

$$(J(\vec{a}_k)^T J(\vec{a}_k))(\vec{a}_{k+1} - \vec{a}_k) = -J(\vec{a}_k)^T \vec{d}(\vec{a}_k),$$

hat man für den $(k+1)$ -ten Iterationsschritt einer Kreisapproximation im \mathbb{R}^2 das lineare Gleichungssystem

$$\begin{pmatrix} J_{d_1, r} & \cdots & J_{d_n, r} \\ J_{d_1, x_0} & \cdots & J_{d_n, x_0} \\ J_{d_1, y_0} & \cdots & J_{d_n, y_0} \end{pmatrix} \begin{pmatrix} J_{d_1, r} & J_{d_1, x_0} & J_{d_1, y_0} \\ \vdots & \vdots & \vdots \\ J_{d_n, r} & J_{d_n, x_0} & J_{d_n, y_0} \end{pmatrix} \begin{pmatrix} r_{k+1} - r_k \\ (x_0)_{k+1} - (x_0)_k \\ (y_0)_{k+1} - (y_0)_k \end{pmatrix} = - \begin{pmatrix} J_{d_1, r} & \cdots & J_{d_n, r} \\ J_{d_1, x_0} & \cdots & J_{d_n, x_0} \\ J_{d_1, y_0} & \cdots & J_{d_n, y_0} \end{pmatrix} \begin{pmatrix} d_1 \\ \vdots \\ d_n \end{pmatrix},$$

wobei $(X_0)_k = \begin{pmatrix} (x_0)_k \\ (y_0)_k \end{pmatrix}$ den Mittelpunkt des Kreises in Schritt k darstellt.

Als Beispiel nimmt Ahn die in Tabelle 5.1 ersichtlichen sechs Punkte im \mathbb{R}^2 .

Tabelle 5.1: Datenpunkte für das Fitten eines Kreises

| | p1 | p2 | p3 | p4 | p5 | p6 |
|---|----|----|----|----|----|----|
| x | 1 | 2 | 5 | 7 | 9 | 3 |
| y | 7 | 6 | 8 | 7 | 5 | 7 |

Um ein Problem der kleinsten Quadrate lösen zu können, benötigt man schließlich noch eine Anfangslösung, von der aus der optimale Kreis berechnet werden sollte. Hier wurde als Anfangswert für den Mittelpunkt der Schwerpunkt der sechs Punkte

$$X_0 = \bar{X} = \frac{1}{n} \sum_{i=0}^n X_i$$

5 Iterative Lösungsverfahren für die Methode der kleinsten Quadrate

und als Radius das quadratische Mittel der Punktabstände zum Schwerpunkt

$$r_0 = \sqrt{\frac{1}{n} \sum_{i=0}^n \|X_i - X_0\|^2}$$

gewählt. Als Lösungsverfahren wurde hier der Gauss-Newton-Algorithmus verwendet.

Nach zehn Iterationsschritten (siehe Abbildung 5.4) bekommt man als Ergebnis einen Kreis mit den Werten

$$X = \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 4.7398 \\ 2.9835 \end{pmatrix} \text{ und } r = 4.7142.$$

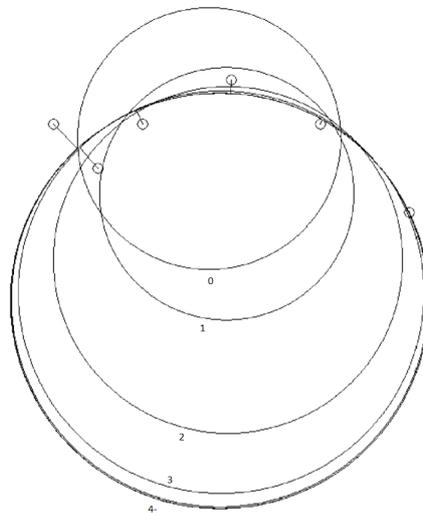


Abbildung 5.4: Iterationsschritte des Kreisfits [3]

Nun wäre es noch interessant, die gegenseitigen Abhängigkeiten der einzelnen Parameter zu betrachten. Hierfür berechnet man die Korrelationsmatrix

$$\text{Cor}(\vec{a}) = \begin{pmatrix} \text{Cor}_{r,r} & \text{Cor}_{r,x} & \text{Cor}_{r,y} \\ \text{Cor}_{x,r} & \text{Cor}_{x,x} & \text{Cor}_{x,y} \\ \text{Cor}_{y,r} & \text{Cor}_{y,x} & \text{Cor}_{y,y} \end{pmatrix} = \begin{pmatrix} 1 & -0.37 & -0.98 \\ -0.37 & 1 & 0.39 \\ -0.98 & 0.39 & 1 \end{pmatrix}$$

und sieht sofort, dass der Radius und die Y-Koordinate des Mittelpunktes des Kreises stark voneinander abhängig sind, was sich auf die Verteilung der Punkte auf der X-Achse zurückführen lässt. Der hohe negative Korrelationswert zwischen dem Radius und der Y-Koordinate bedeutet also, dass die Y-Koordinate des Mittelpunktes fällt, wenn der Radius steigt, und umgekehrt. Dies lässt sich auch in der Abbildung 5.4 gut erkennen.

5 Iterative Lösungsverfahren für die Methode der kleinsten Quadrate

Die Korrelationen, die bei dem gegebenen Datensatz ersichtlich sind, sind natürlich keine allgemein gültigen Gesetze. Die Ergebnisse variieren je nach Beschaffenheit des gegebenen Datensatzes.

6 Ein Anwendungsbeispiel

Zum Abschluss dieser Arbeit betrachten wir noch ein Anwendungsbeispiel aus dem Bereich der Messtechnik.

Werkzeughersteller müssen für ihre Produkte gewisse Standards beziehungsweise Normen einhalten, um für die Herstellungsqualität garantieren zu können. Darunter zählen neben Form des Werkzeugs auch die Haltbarkeit beziehungsweise der Verschleiß, wenn dieses Produkt für längere Zeit unter Verwendung ist.

In diesem Abschnitt betrachten wir Werkzeuge mit Schneidkanten, wie zum Beispiel Bohrer oder Fräser, und geben ein Beispiel dafür, wie man die Form einer solchen Schneidkante mittels der Methode der kleinsten Quadrate analysieren kann.

Zu Beginn benötigt man eine Punktmenge, die diese Kante repräsentiert. Hierfür wurden einige Messsysteme von Firmen entwickelt, die sich darauf spezialisiert haben, solche Arten von Werkzeug zu messen und mittels einer 3D-Punktwolke darzustellen.

Da es in diesem Beispiel um die durchschnittliche Form einer Schneidkante geht, reicht es, 2D-Profile, die über die Kante gelegt werden, aus dieser Punktwolke zu extrahieren und diese entweder zu mitteln, um ein Profil zu bekommen, das repräsentativ für einen Teil des Werkzeugs ist, oder auch die Profile einzeln zu betrachten und zu analysieren. Der Vorgang der Extraktion ist für dieses Beispiel jedoch nicht relevant und wird somit hier nicht weiter erklärt.

Sei nun also $\{X_i\}_{1 \leq i \leq n}$ die Menge an Punkten im \mathbb{R}^2 , die das zu betrachtende Profil darstellen. In Abbildung 6.1 sieht man eine Aufteilung eines solchen Profils in vier Abschnitte, die für das Analysieren der Form des Werkzeuges benötigt werden.

Die gegebene Kante wird also eingeteilt in:

- Freifläche,
- Fase,
- Schneidenradius und
- Spanfläche,

wobei dies natürlich auch gespiegelt sein kann.

Nun müssen folgende zwei Fragen beantwortet werden:

1. „Was soll analysiert werden?“ und
2. „Wie soll es analysiert werden?“.

6 Ein Anwendungsbeispiel

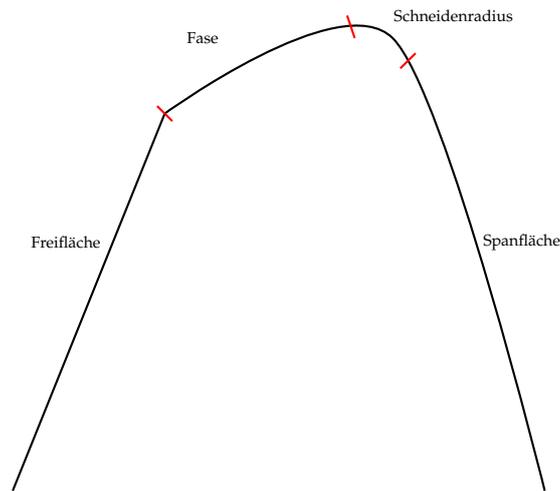


Abbildung 6.1: Einteilung eines Profils in vier Abschnitte

Die Frage nach dem „Was“ lässt sich schnell beantworten.

Um die Qualität solcher Schneidkanten zu überprüfen, interessieren sich die Hersteller besonders für den Radius des gleichnamigen Bereiches des Schneidenradius, die Länge der Fase und auch für die Winkel zwischen den einzelnen Flächen.

Nun zu der Frage nach dem „Wie“:

Wir benötigen in diesem Schritt geometrische Formen, mit denen man diese Punktmenge approximieren kann, um die gewünschten Ergebnisse zu bekommen. Man erkennt sofort, dass man das Profil nicht mittels eines einzelnen einfachen geometrischen Objektes darstellen kann, aus dem sich die gesuchten Parameter herauslesen lassen. Somit sucht man nach einer Aufteilung der gegebenen Punktmenge und einem geometrischen Objekt pro Aufteilung, mittels dem approximiert werden soll, und mit Hilfe derer man die gewünschten Parameter berechnen kann.

Hier wird die schon beschriebene Aufteilung in die vier Bereiche Freifläche, Fase, Schneidenradius und Spanfläche (siehe Abbildung 6.1) gewählt. Um diese vier Bereiche zu approximieren, werden folgende Objekte gewählt:

- Die Freifläche, die Fase und die Spanfläche sollen mit Hilfe einer Geraden approximiert werden.
- In den Bereich des Schneidenradius soll ein Kreissegment gefittet werden.

Das Approximieren mittels Geraden wird aus dem Grund gewählt, da dies einerseits der Form der drei Abschnitte gut entspricht, und sich dadurch andererseits die gesuchten Winkel zwischen den Bereichen schnell berechnen lassen. Anstelle des Kreises für die Approximation des Schneidenradius könnte man ebenso eine Ellipse wählen.

6 Ein Anwendungsbeispiel

Bis jetzt wurden nur Beispiele betrachtet, in denen Punktmengen mittels eines einzelnen Objekts approximiert wurden, jedoch lässt sich dies natürlich auf eine beliebige Anzahl erweitern. Hierfür muss jedoch die Zugehörigkeit der einzelnen Punkte zu den Abschnitten geklärt sein.

Sei also $S_0 = \{L_{FF}, L_F, C, L_{SF}\}$ eine Anfangslösung, wobei die einzelnen Parameter die approximierenden Objekte darstellen, das bedeutet:

- $L_{FF} \dots$ eine Gerade, die im Bereich der Freifläche liegt
- $L_F \dots$ eine Gerade, die im Bereich der Fase liegt
- $C \dots$ ein Kreis, der im Bereich des Schneidenradius liegt
- $L_{SF} \dots$ eine Gerade, die im Bereich der Spanfläche liegt

Die einzelnen Objekte der Anfangslösung sollten natürlich relativ gut in den zugehörigen Abschnitten liegen, damit die in den weiteren Iterationen der Methode der kleinsten Quadrate bestimmten Zuordnungen der Punkte zu den Objekten eine vernünftige Lösung liefern können.

Ein wichtiger Punkt, der für die Approximation beachtet werden muss, ist das Zusammenhängen der einzelnen Objekte. Es sollte ein nahtloser Übergang von der Geraden L_F der Fase zum Kreissegment C und von der Geraden L_{SF} der Spanfläche zum Kreissegment C bestehen, da ein Abstand zwischen diesen Objekten zu keinem realistischen Ergebnis führen würde (siehe Abbildung 6.2). Diese Eigenschaft sollte man auch schon in die Anfangslösung miteinfließen lassen.

Nun zur Zuordnung der Punkte:

Zuerst betrachten wir den Übergang der Geraden L_F zum Kreissegment C und dann weiter zur Geraden L_{SF} . Die zwei Geraden müssen tangential auf dem eingepassten Kreis liegen, wobei die Endpunkte beider Geraden gleich ihres Tangentenpunktes am Kreis sind.

Ein Punkt, der also über den Objekten liegt, lässt sich leicht mittels orthogonalem Abstand zuordnen (siehe Abbildung 6.3). Für einen Punkt unterhalb der Objekte muss überprüft werden, ob dieser im Bereich zwischen dem Kreismittelpunkt, den Tangentenpunkten der zwei Geraden und dem Kreissegment liegt, und somit dem Kreis zugeordnet werden muss, oder außerhalb dieses Bereiches, und somit zu einer der zwei Geraden gehört (siehe Abbildung 6.5).

Für die drei Randbereiche

- Punkte auf der Normalen zur Geraden L_F im Berührungspunkt,
- Punkte auf der Normalen zur Geraden L_{SF} im Berührungspunkt und
- Punkte auf der Winkelhalbierenden der Geraden L_F und L_{SF} , die nicht dem Kreissegment zugeordnet sind,

6 Ein Anwendungsbeispiel

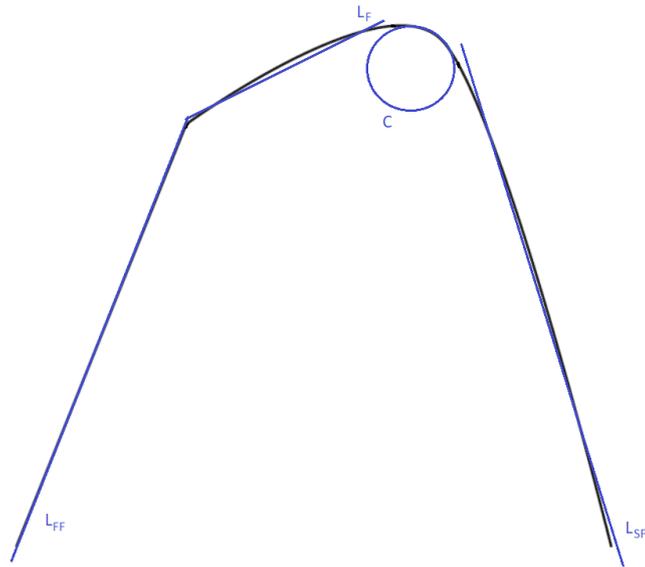


Abbildung 6.2: Ein unrealistisches Ergebnis der Approximation

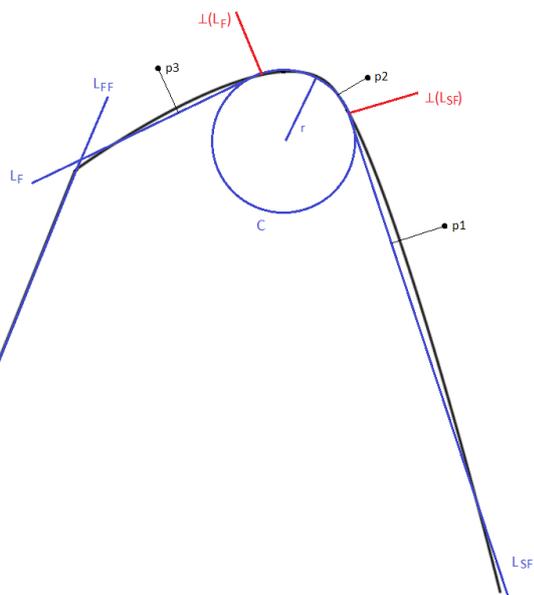


Abbildung 6.3: Zuordnung der Punkte für die obere rechte Seite des Profils

6 Ein Anwendungsbeispiel

Eingabe: Ein Datenpunkt X , die Geraden L_F , L_{FF} und L_{SF} und der Kreis C

Ausgabe: Der Bereich des Profils mit minimalem Abstand zum Punkt X

Initialisierung $Bereich = unbekannt$

```
if  $X$  liegt zwischen der Normalen auf die Gerade  $L_F$  im Berührungspunkt mit dem Kreis  $C$  und der
Normalen auf die Gerade  $L_{SF}$  im Berührungspunkt mit dem Kreis  $C$  then
|  $Bereich = C$ 
end
else
| if  $X$  liegt auf der rechten Seite der Winkelhalbierenden der Geraden  $L_F$  und  $L_{SF}$  und auf
| der rechten Seite der Winkelhalbierenden der Geraden  $L_{FF}$  und  $L_{SF}$  then
| |  $Bereich = L_{SF}$ 
| end
| else
| | if  $X$  liegt auf der linken Seite der Winkelhalbierenden der Geraden  $L_F$  und  $L_{FF}$  then
| | |  $Bereich = L_{FF}$ 
| | end
| | else
| | |  $Bereich = L_F$ 
| | end
| end
end
return  $Bereich$ 
```

Abbildung 6.4: Zuordnung eines Punktes

muss die Zuordnung individuell festgelegt werden, da hier der Abstand zu je zwei Bereichen übereinstimmt.

Betrachten wir nun den Übergang der Gerade L_F zur Gerade L_{FF} . Für Punkte, die unter den zwei Geraden liegen, lässt sich leicht mittels orthogonalem Abstand entscheiden, zu welchem Objekt der jeweilige Punkt zugeordnet werden soll (siehe Abbildung 6.5). Für Punkte im oberen Bereich muss schließlich mittels der Winkelhalbierenden der zwei Geraden entschieden werden, da der Vergleich des orthogonalen Abstandes zu falschen Ergebnissen führen könnte (siehe Abbildung 6.6). Für die drei Randbereiche

- Punkte auf der Normalen zur Geraden L_F im Schnittpunkt der zwei Geraden,
- Punkte auf der Normalen zur Geraden L_{FF} im Schnittpunkt der zwei Geraden und
- Punkte auf der Winkelhalbierenden der Geraden L_F und L_{FF} ,

muss die Zuordnung wieder individuell festgelegt werden.

Als Zusammenfassung ergibt sich der in Abbildung 6.4 ersichtliche Algorithmus.

Nachdem die Zuordnung der Punkte geklärt ist, muss noch entschieden werden, welche Objektparameter des Optimieralgorithmus verändert werden dürfen. Hierfür gibt es

6 Ein Anwendungsbeispiel

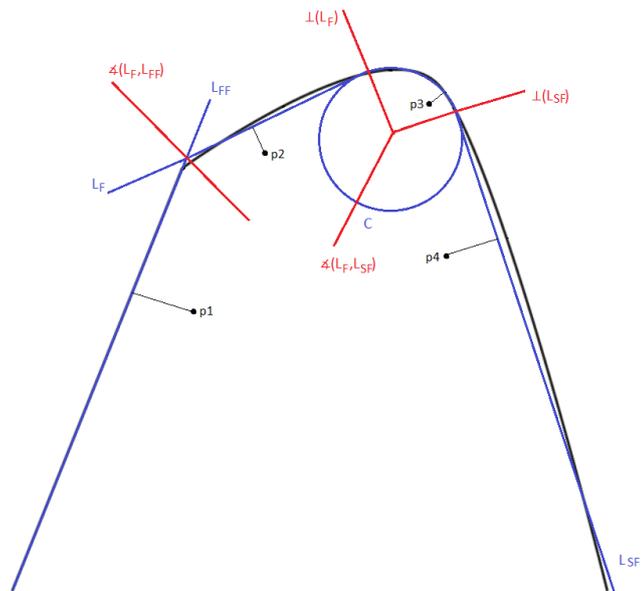


Abbildung 6.5: Zuordnung der Punkte für die untere Seite des Profils

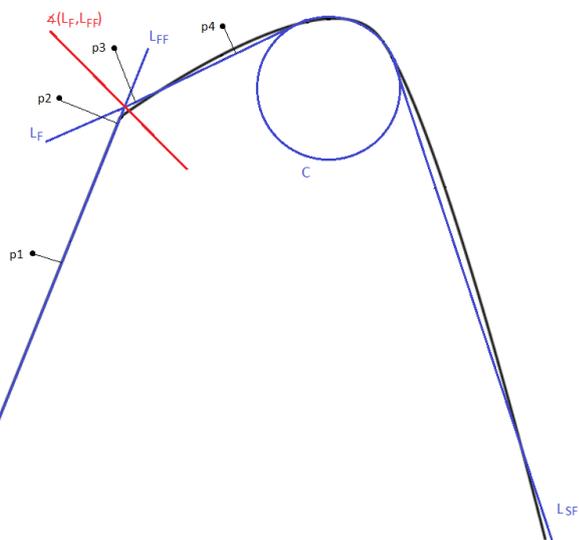


Abbildung 6.6: Zuordnung der Punkte für die obere linke Seite des Profils

6 Ein Anwendungsbeispiel

einige Vorgaben:

1. Die drei Geraden L_F , L_{FF} und L_{SF} sollten sowohl Winkel als auch Position ändern können.
2. Die Länge der Gerade L_F sollte variabel sein.
3. Das Kreissegment sollte sowohl Radius als auch Position ändern können.
4. Die einzelnen Objekte sollten wie oben beschrieben zusammenhängen.

Aufgrund der Bedingung in Punkt 4 müssen die einzelnen Positionen der Geraden und des Kreissegments voneinander abhängig sein, wodurch nahe gelegt ist, dass der Optimieralgorithmus nicht die Position aller Objekte beeinflussen sollte, sondern nur die notwendigsten. Wir beschränken uns hier auf die Veränderung der Positionen der Geraden L_{FF} und L_{SF} , da sich die Position der anderen Objekte daraus ergibt. Hier könnte jedoch ebenfalls die Position des Kreises und der Gerade L_F als beeinflussbare Objektparameter gewählt werden.

Aus Punkt 1 und 2 wird ersichtlich, dass auch die Winkel der drei Geraden und der Radius des Kreissegments als variable Objektparameter übergeben werden müssen, um dem Algorithmus die gewünschten Freiheiten im Fitten ermöglichen zu können.

Sollte man sich einen fixen Winkel einer Geraden wünschen, kann man den zugehörigen Objektparameter auch weglassen, und die Berechnung der Punktezuordnung und -distanzen dementsprechend verändern. Das selbe gilt für den Radius des Kreissegments.

Bemerkung 6.1. Möchte man also eine Approximation mit einem Objekt, der einen fixen Parameter besitzen soll (zum Beispiel der Radius eines Kreises), durchführen, gibt man den entsprechenden Objektparameter nicht an den Optimierer weiter und inkludiert ihn einfach fix in die Berechnung der Punktezuordnung (bei mehreren Objekten) und der Distanzberechnung.

Somit ergeben sich als variable Objektparameter

- der Polarwinkel und die Länge der Geraden L_F ,
- die Position der Geraden L_{FF} und L_{SF} und
- der Radius des Kreissegmentes C .

Nachdem die Frage der Parametrisierung geklärt ist, muss noch definiert werden, mit welchem Algorithmus dieses Problem gelöst werden soll.

Hier wird die Umsetzung mit der Levenberg-Marquardt Methode vorgeschlagen, da diese auf Grund der variablen Schrittweite gut auf Verschlechterungen in der Fehlerfunktion reagieren kann und je nach Verfahren zur Lösung der linearen Gleichungssysteme sehr stabil sein kann. [28]

Der Levenberg-Marquardt-Algorithmus wurde in C++ umgesetzt und mit der oben beschriebenen Parametrisierung und Punktezuordnung verknüpft. Für die Auswertung wurden mehrere Schneidkanten betrachtet, von denen drei hier genauer beschrieben

6 Ein Anwendungsbeispiel

werden. Diese drei Kanten wurden auf Grund der unterschiedlichen Größen der approximierenden Kreise als auch auf Grund der unterschiedlichen Länge der Geraden L_F gewählt. Man sieht am Ergebnis (siehe Tabelle 6.2), dass der Algorithmus trotz der großen Unterschiede der Kanten immer eine Lösung findet.

Für jede Schneidkante wurden Profile extrahiert und deren Mittel schließlich an den Algorithmus übergeben. Die Anzahl der Punkte pro Profil ist in Tabelle 6.1 ersichtlich.

Tabelle 6.1: Punktzahl der Profile der betrachteten Kanten

| | Punktzahl |
|---------|-----------|
| Kante 1 | 718 |
| Kante 2 | 2698 |
| Kante 3 | 1123 |

Kommen wir nun zu den Ergebnissen der Approximation:

Mittels der optimalen Parameter, die der Levenberg-Marquardt-Algorithmus ermittelt, wurden zusätzlich zum Radius des Kreises C noch die Länge zwischen Berührungspunkt der Geraden L_F mit dem Kreis und Schnittpunkt von L_F mit der Geraden L_{FF} und auch der Winkel zwischen den Geraden L_{FF} und L_{SF} berechnet.

Jede der hier beschriebenen Kanten wurde 50 Mal ausgewertet, um die Stabilität des Algorithmus zu überprüfen. Aufgrund der Abbruchbedingungen und der variablen Schrittweite des Levenberg-Marquardt-Verfahrens lässt sich eine kleine Schwankung in den Ergebnissen erkennen.

Tabelle 6.2: Ergebnisse der Approximation für drei Schneidkanten

| | Kante 1 | |
|----------------------------------|------------|--------------------|
| | Mittelwert | Standardabweichung |
| Radius von C | 6.515 | 0.012 |
| Länge von L_F | 68.509 | 0.013 |
| Winkel zw. L_{FF} und L_{SF} | 65.411 | 0.003 |
| | Kante 2 | |
| | Mittelwert | Standardabweichung |
| Radius von C | 6.012 | 0.008 |
| Länge von L_F | 13.879 | 0.011 |
| Winkel zw. L_{FF} und L_{SF} | 89.588 | 0.011 |
| | Kante 3 | |
| | Mittelwert | Standardabweichung |
| Radius von C | 40.783 | 0.002 |
| Länge von L_F | 65.639 | 0.011 |
| Winkel zw. L_{FF} und L_{SF} | 79.111 | 0.003 |

6 Ein Anwendungsbeispiel

Betrachtet man Tabelle 6.2, werden hier die Mittelwerte über alle Auswertungen pro Kante aufgezeigt. Die Standardabweichungen beschreiben die Schwankung der Ergebnisse des Algorithmus.

Sieht man sich die Ergebniswerte der drei Kanten genauer an, sieht man, dass der Wert für die Standardabweichung des Radius und des Winkels zwischen den Geraden L_{FF} und L_{SF} besonders für die Kante 3 sehr gering ist. Dies lässt sich dadurch erklären, dass auf Grund der Größe des Bereiches des Schneidkantenradius der Kante 3 viel mehr Punkte für die Approximation des Kreises C vorhanden sind, und es für den Algorithmus somit leichter ist, eine stabile Lösung zu ermitteln (siehe Abbildung 6.9).

Die Standardabweichung für den Winkel zwischen den Geraden L_{FF} und L_{SF} ist für die Kante 2 um einiges höher als für die anderen hier betrachteten. Der Grund hierfür konnte jedoch nicht herausgefunden werden.

Die Darstellung des Ergebnisses für alle drei Kanten für je einen Durchlauf ist in Abbildung 6.7 für Kante 1, in Abbildung 6.8 für Kante 2 und in Abbildung 6.9 für Kante 3 ersichtlich. In blau sind die für die Approximation verwendeten Punkte des Profils eingezeichnet. Die Gerade L_F wurde der Anschaulichkeit halber bis zu ihrem Schnittpunkt mit der Geraden L_{SF} verlängert.

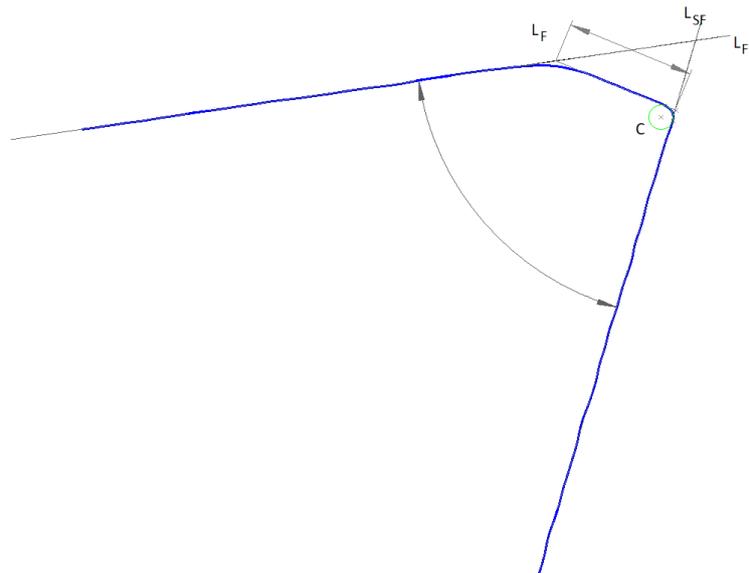


Abbildung 6.7: Ergebnis der Approximation für Kante 1

6 Ein Anwendungsbeispiel

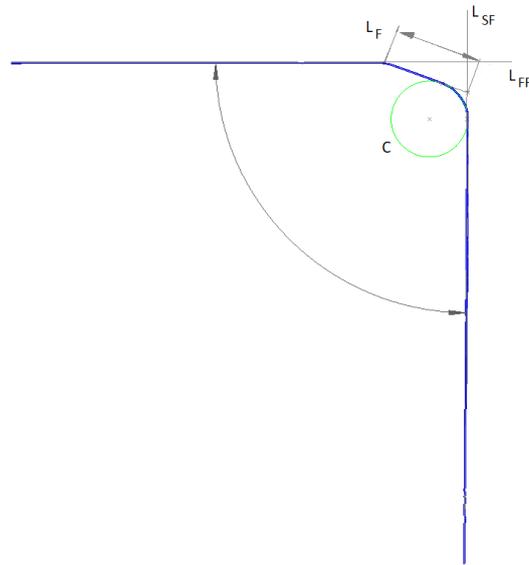


Abbildung 6.8: Ergebnis der Approximation für Kante 2

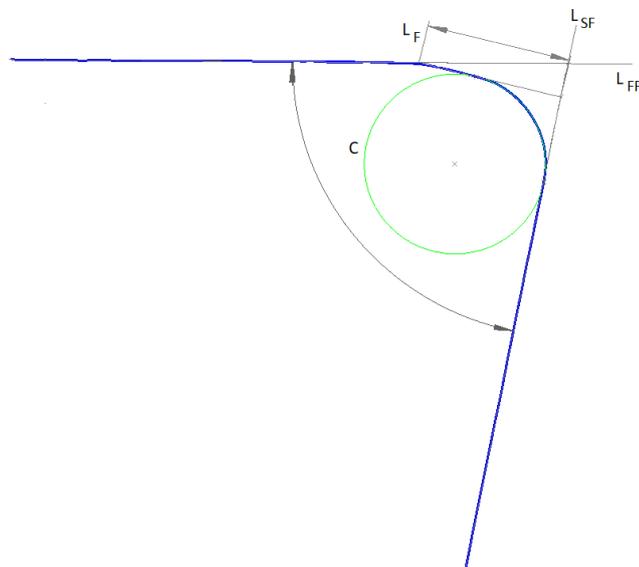


Abbildung 6.9: Ergebnis der Approximation für Kante 3

7 Fazit und Ausblick

Im Rahmen dieser Arbeit wurde das Approximieren von Punktmengen mit einfachen geometrischen Objekten mittels des minimalen orthogonalen Abstandes betrachtet.

Es wurde der Zusammenhang zwischen dem Verfahren der Hauptkomponentenanalyse, die hauptsächlich im Bereich der Statistik zum Reduzieren der Anzahl von Dimensionen eines Datensatzes verwendet wird, und dem Approximieren von Geraden und Ebenen in eine Menge von Datenpunkten hergestellt, der auf dem Bezug zwischen dem Minimieren der Summe der kürzesten orthogonalen Distanzen und dem Maximieren der Varianz der Daten beruht.

Diese Methode der Approximation besitzt jedoch den Nachteil, sehr sensibel auf Ausreißerdaten zu reagieren, weshalb weitere robuste Varianten der Hauptkomponentenanalyse entwickelt wurden, die besser mit solchen Punktmengen umgehen können.

Der Begriff „Robuste Hauptkomponentenanalyse“ ist breit gefächert und versucht die verschiedensten Aspekte der Hauptkomponentenanalyse zu verändern beziehungsweise zu verbessern. Auf Grund der großen Anzahl von robusten Algorithmen wurde in dieser Arbeit versucht, zwei dieser Methoden, die unterschiedliche Aspekte der Hauptkomponentenanalyse bearbeiten, vorzustellen.

Die Hauptkomponentenanalyse mit Gewichtsiterationen ist eine der bekannteren Methoden, wobei ihr Erfolg sehr stark von der gewählten Gewichtsfunktion abhängt. In unserem Beispiel wurde eine Gewichtsfunktion gewählt, die je nach Parametrisierung sehr schnell zu einem guten Ergebnis führen kann.

Im Folgenden wurden weitere iterative Verfahren behandelt, die zum Approximieren mittels geometrischen Objekten verwendet werden können. Als Anwendungsbeispiel aus dem Bereich der Messtechnik wurden Profile einer Schneidkante betrachtet, die mit verschiedenen geometrischen Objekten approximiert wurden. Zur Lösung dieser Problemstellung wurde auf das Levenberg-Marquardt-Verfahren zurückgegriffen, da dieses auf Grund der variablen Schrittweite sehr robust und schnell zum Ziel führt.

Hier würde noch die Möglichkeit bestehen, die selbe Problemstellung mittels eines anderen iterativen Verfahrens, wie zum Beispiel des Gauss-Newton-Verfahrens, zu lösen und die Ergebnisse sowie die Dauer der Auswertung zu vergleichen.

7 Fazit und Ausblick

Auch die Wahl der zu approximierenden Objekte könnte verändert und gegenübergestellt werden. Hier würden sich zum Beispiel NURBS-Funktionen anbieten, da diese dem Verlauf der Kante besser folgen könnten.

Literaturverzeichnis

- [1] DIN 32880-1. *Coordinate Metrology. Geometrical Fundamental Principles, Terms and Definitions*. Beuth Verlag Berlin, 1986.
- [2] S.J. Ahn. *Least Squares Orthogonal Distance Fitting of Curves and Surfaces in Space*. Springer-Verlag Berlin Heidelberg, 1998.
- [3] S.J. Ahn, W. Rauh, and H.-J. Warnecke. Least-squares orthogonal distances fitting of circle, sphere, ellipse, hyperbola, and parabola. *Pattern Recognition*, Volume 34:2283–2303, 2001.
- [4] D. Bates. Matrix decompositions for regression analysis. <https://www.stat.wisc.edu/courses/st849-bates/lectures/Orthogonal.pdf>, 2010.
- [5] R. Bhatia. *Matrix Analysis*. Springer-Verlag, 1997.
- [6] I. N. Bronštejn, K. A. Semendjajew, G. Musiol, and H. Mühlig. *Taschenbuch der Mathematik*. Verlag Harri Deutsch, Thun, 1995.
- [7] J. Cadima and I. Jolliffe. On relationships between uncentered and column-centered principal component analysis. *Pakistan Journal of Statistics*, Volume 25(4):473–503, 2009.
- [8] N. Chernov and C. Lesort. *Least squares fitting of circles and lines*. Department of Mathematics, University of Alabama at Birmingham, 2003.
- [9] J. W. Demmel. *Applied Numerical Linear Algebra*. Society for Industrial and Applied Mathematics Philadelphia, 1997.
- [10] G. Farin, J. Hoschek, and M.-S. Kim. *Handbook of Computer Aided Geometric Design*. Elsevier Science, Amsterdam, 2002.
- [11] A. Fitzgibbon, M. Pilu, and R. B. Fisher. Direct least square fitting of ellipses. *Pattern Analysis And Machine Intelligence*, Volume 21(5):476–480, 1999.
- [12] J. E. Gentle. *Matrix Algebra: Theory, Computations and Applications in Statistics*. Springer-Verlag, 2007.
- [13] K. Hauser. Multivariate Newton’s method and Quasi-Newton methods. http://homes.soic.indiana.edu/classes/spring2012/csci/b553-hauserk/newtons_method.pdf, 2012. Stand 22.06.2018.

- [14] J. Jauregui. Principal component analysis with linear algebra. <http://www.math.union.edu/~jaureguj/PCA.pdf>, 2012. Stand 22.06.2018.
- [15] S. L. Keeling and K. Kunisch. Robust l_1 approaches to computing the geometric median and principal and independent components. *Journal of Mathematical Imaging and Vision*, Vol. 56 (1):99–124, 2014.
- [16] R. Klein. *Algorithmische Geometrie*. Springer-Verlag Berlin Heidelberg, 2005.
- [17] D. Q. Lee. Numerically efficient methods for solving least squares problems. <http://www.math.uchicago.edu/~may/REU2012/REUPapers/Lee.pdf>, 2012.
- [18] K. Levenberg. A method for the solution of certain nonlinear problems in least squares. *Quarterly of Applied Mathematics*, Volume 2:164–168, 1944.
- [19] M. Lira, R. Iyer, A. A. Trindade, and V. Howle. QR versus Cholesky: A probabilistic analysis. *International Journal Of Numerical Analysis And Modeling*, Volume 13:114–121, 2016.
- [20] C.F. Van Loan. *Introduction to Scientific Computing: A Matrix-vector Approach Using MATLAB*. Prentice Hall, 1997.
- [21] K. Madsen, H. B. Nielsen, and O. Tingleff. *Methods for Non-Linear Least Squares Problems (2nd ed.)*. Informatics and Mathematical Modelling, Technical University of Denmark, 2004.
- [22] D. W. Marquardt. An algorithm for least squares estimation of nonlinear parameters. *SIAM J. Appl. Math.*, Volume 11:431–441, 1963.
- [23] K. Pearson. On lines and planes of closest fit to systems of points in space. *The Philosophical Magazine*, Vol. 6 (2):559–572, 1901.
- [24] T. Richter and T. Wick. *Einführung in die Numerische Mathematik: Begriffe, Konzepte und zahlreiche Anwendungsbeispiele*. Springer Berlin Heidelberg, 2017.
- [25] C. M. Shakarji and V. Srinivasan. Theory and algorithms for weighted total least-squares fitting of lines, planes, and parallel planes to support tolerancing standards. *Journal of Computing and Information Science in Engineering*, Volume 13(3):031008–031008–11, 2013.
- [26] L.I. Smith. A tutorial on principal components analysis. Technical report, University of Otago Library, 05 2002. http://www.iro.umontreal.ca/~pift6080/H09/documents/papers/pca_tutorial.pdf.
- [27] D. H. von Seggern. *CRC standard curves and surfaces*. CRC Press, Boca Raton, 1993.
- [28] D. Weisskopf. *Untersuchung, Implementierung und Einsatz des Levenberg-Marquardt-Algorithmus zur Analyse von Chip-Fertigungsdaten*. Diplomarbeiten Agentur diplom.de, 1999.