

Untersuchung einer inversen Abbildung zu nichtlinearen Dimensionsreduktionen mit Anwendung auf Simulationsdaten

Sara Hahner

Geboren am 7. Januar 1995 in Fulda

9. September 2016

Bachelorarbeit Mathematik

Betreuer: Prof. Dr. Jochen Garcke

Zweitgutachter: Prof. Dr. Marc Alexander Schweitzer

INSTITUT FÜR NUMERISCHE SIMULATION

Betreuer am Fraunhofer SCAI: Rodrigo Iza-Teran

FRAUNHOFER-INSTITUT FÜR ALGORITHMEN UND WISSENSCHAFTLICHES RECHNEN

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

Inhaltsverzeichnis

Notations- und Abkürzungsverzeichnis	iii
1. Einleitung	1
2. Dimensionsreduktion	5
2.1. Lineare Dimensionsreduktion	7
2.2. Nichtlineare Dimensionsreduktion	9
2.2.1. Isomap	9
2.2.2. Locally Linear Embedding	13
2.2.3. Diffusion Maps	15
3. Die Rückabbildung	19
3.1. Funktionsweise	19
3.2. Zur Lösung des Optimierungsproblems	23
3.3. Aufwand	25
4. Anwendungen	27
4.1. Theoretische Aspekte der Analyse	30
4.2. Untersuchung	31
4.2.1. Dimensionsreduktion	32
4.2.2. Transformation	33
4.2.3. Inversion der Dimensionsreduktion mit LLI	35
4.2.4. Vollständige Rückabbildung	35
4.2.5. Variation der Nachbarschaftswahl	38
4.2.6. Laufzeitvergleich	42
4.3. Ergebnisse	44
5. Resümee und Ausblick	47
A. Plots der Dimensionsreduktionen	49
Abbildungsverzeichnis	57
Literatur	59

Notation

$\mathbb{N}$	Die Menge der natürlichen Zahlen: $\{0, 1, 2, 3, \dots\}$
$\mathbb{R}$	Die Menge der reellen Zahlen
I_p	$p \times p$ Einheitsmatrix
$\mathbb{1}$	Spaltenvektor mit Einsen $(1, \dots, 1)^T$
N	Anzahl der betrachteten Datenpunkte
$\mathcal{P}$	Die Menge der möglichen Parametereinstellungen im $\mathbb{R}^r$
Ω	Die Menge der betrachteten Parametereinstellungen: $\{p^1, \dots, p^N\} \subset \mathcal{P} \subseteq \mathbb{R}^r$
$\mathcal{X}$	Die Menge der betrachteten hochdimensionalen Datenpunkte: $\{x^1, \dots, x^N\} \subset \mathbb{R}^d$
$\mathcal{X}_{red}$	Die eingebettete, niedrigdimensionale Menge der Datenpunkte: $\{y^1, \dots, y^N\} \subset \mathbb{R}^p$
r	Dimension von $\mathcal{P}$
d	Dimension von $\mathcal{X}$
p	Dimension von $\mathcal{X}_{red}$
$\mathcal{I}$	Die Menge der Indizes der k nächsten Nachbarn in $\mathcal{X}_{red}$
$\mathcal{I}_{\mathcal{X}}$	Die Menge der Indizes der k nächsten Nachbarn in $\mathcal{X}$
$\mathcal{I}_{\mathcal{P}}$	Die Menge der Indizes der k nächsten Nachbarn in $\mathcal{P}$
p^*	Die Parametereinstellung, deren Simulation approximiert wird
$\hat{y}$	Die ursprüngliche Einbettung in $\mathcal{X}_{red}$ zur Parametereinstellung p^*
y^*	Das approximierte Bild der Parametereinstellung p^* im Einbettungsraum
x^*	Die Approximation von $f_{sim}(p^*)$ im hochdimensionalen Raum

Abkürzungen

SVD	Singulärwertzerlegung (Singular value decomposition)
PCA	Hauptachsentransformation (Principal component analysis)
CMDS	Klassische Multidimensionale Skalierung (Classical multidimensional scaling)
EVD	Eigenwertzerlegung (Eigenvalue decomposition)
Diff. Maps	Diffusion Maps
LLE	Locally Linear Embedding
RBF	Radiale Basisfunktion
LLI	Lineare lokale Interpolation (2. Teil der Rückabbildung)
u.d.N.	unter der Nebenbedingung

1. Einleitung

In der Entwicklung von Produkten spielen computergestützte Simulationen eine sehr große Rolle, da sie einfachere, schnellere und kostengünstigere Untersuchungen von Systemen ermöglichen.

Ein bekanntes Beispiel sind Autocrashtests, mittels derer versucht wird, den Einfluss der Werkstoffe und der Konstruktion auf die Personensicherheit zu verstehen. Die Tests werden für verschiedene Parametereinstellungen simuliert, welche Eingaben bei den Simulationen sind, wie zum Beispiel die Zusammensetzung des Metalls, seine Dicke oder Verarbeitung, und die Ergebnisse analysiert.

Des Weiteren wird das Verhalten einzelner Werkstücke bei Zugversuchen betrachtet, in deren Rahmen die Auswirkungen von Verformungen auf Metallplatten untersucht werden. Nun werden die korrespondierenden Spannungs-Dehnungs-Diagramme (siehe Abbildung 1.1a) in Abhängigkeit verschiedener Eigenschaften des Materials ausgewertet.

Bei oben beschriebenen Simulationen wird ein feines Gitter aus Punkten über das Werkstück gelegt und die Verschiebungen der Gitterpunkte werden auf Basis physikalischer Zusammenhänge approximiert. Die Ergebnisse der Simulation können die Positionen der einzelnen Gitterpunkte zu verschiedenen Zeitpunkten sein, aber auch Diagramme, welche weitere interessante Zusammenhänge zeigen.

Die Ergebnisse enthalten im Allgemeinen (zu) viele Merkmale, sind somit hochdimensional und unhandlich, wodurch eine vergleichende Analyse erschwert wird. Glücklicherweise sind viele Informationen redundant, weil das Verhalten verschiedener Gitterpunkte in hohem Maße voneinander abhängt. Deshalb wird zur Untersuchung von Simulationsdaten häufig eine *Dimensionsreduktion* verwendet, welche unwichtige Informationen reduziert und damit die Dimensionalität verringert. So können im niedrigdimensionalen Raum das Verhalten des simulierten Prozesses analysiert und Abhängigkeiten von Parametereinstellungen abgelesen werden, wie es in Abbildung 1.1b der Fall ist.

Computersimulationen sind sehr komplex, weshalb versucht wird, ihre Anzahl klein zu halten. Stattdessen werden Ergebnisse für neue Bedingungen unter Verwendung schon vorhandener Daten approximiert, um Auffälligkeiten aufzudecken, die eine gezielte Auswahl von Parametereingaben für die Simulation ermöglichen. (Liegen beispielsweise im Kontext der Autocrashtests zahlreiche, gegebenenfalls approximierte, Simulationsergebnisse vor, offenbart sich, dass geringe Blechdicke unabhängig der restlichen gewählten Parametereinstellungen zum Versagen des Systems führt.)

Vorstehende Approximationen werden mittels Interpolation zwischen Simulationsergebnissen gewonnen. Diese ist jedoch häufig teuer und kompliziert, da die intrinsische Struktur der Daten nicht unmittelbar ausgenutzt wird.

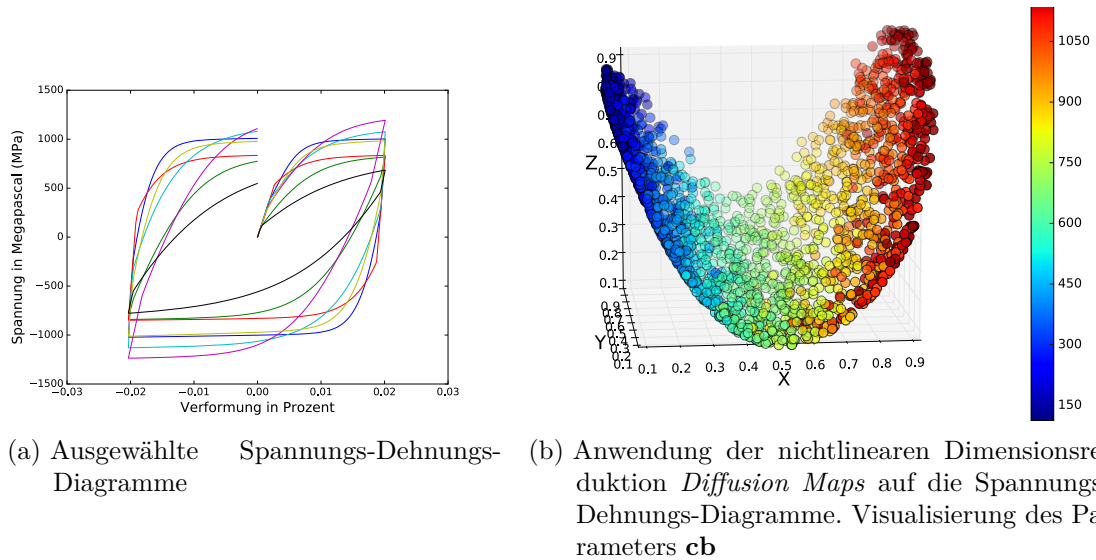


Abbildung 1.1.: Dimensionsreduktion des Beispiels ZUGVERSUCH

Problemstellung

Bei Dimensionsreduktionen werden genau solche intrinsischen Eigenschaften der Simulationsergebnisse hervorgehoben und sollen nun als Grundlage zur Interpolation der Simulationsergebnisse dienen.

Anstatt direkt aus dem Parameterraum in den Raum der Simulationen zu interpolieren, wird der bei der Dimensionsreduktion entstandene Einbettungsraum benutzt. Interpoliert wird somit in zwei Schritten (siehe Abbildung 1.2): Zu einer gewünschten Parametereinstellung wird zunächst die Einbettung der dazugehörigen, unbekannten Simulation angenähert. Im Anschluss wird eine Inversion der Dimensionsreduktion auf diese Approximation angewendet, um das gesuchte Ergebnis vorherzusagen.

Von diesem Ansatz erhofft man sich im Vergleich zu einer direkten Interpolation genauere Resultate bei geringerem Aufwand. Im Falle von Isomap, einem Algorithmus zur nichtlinearen Dimensionsreduktion, wurde in [Fra15] und [Fra+14] eine solche Vorgehensweise erfolgreich auf Simulationen der transsonischen Strömungen bei Flugzeugen angewandt. Dieses Verfahren erhielt den Namen *Backmapping* (Rückabbildung).

Diese Erörterung stellt den Versuch dar, eine Analyse der Funktionalität der Rückabbildung zu erarbeiten. Deshalb wird der Algorithmus auf die mir vorliegenden Simulationen angewendet, um die Ergebnisse zu vergleichen und fallbezogen Verbesserungen am Algorithmus vorzunehmen.

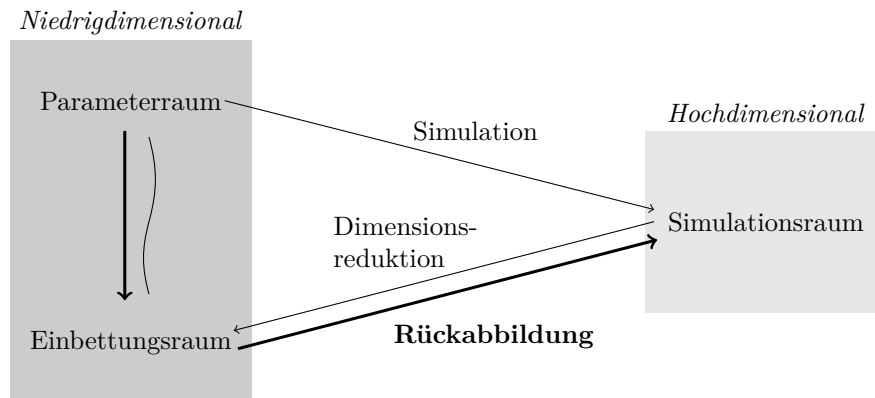


Abbildung 1.2.: Mengen und Abbildungen bei der Rückabbildung

Eigener Anteil und Weiterentwicklungen

- Aufarbeitung der in [Fra15] und [Fra+14] eingeführten Vorgehensweise zur Approximation von Simulationsergebnissen
- Implementierung des Verfahrens zur Inversion von Isomap
- Anwendung der Methode auf weitere nichtlineare Dimensionsreduktionen
- Implementierung verschiedener Variationen der gesamten Rückabbildung zur Approximation von Simulationsergebnissen
- Analyse der Effektivität und Effizienz des Algorithmus an verschiedenen synthetischen und realen Datensätzen
- Anwendungsbezogene Weiterentwicklung der Rückabbildung im Falle schlechter Vorhersage eingebetteter Datenpunkte

Aufbau der Arbeit

Zuerst werden in Kapitel 2 grundlegende Informationen zu den Funktionsweisen der verwendeten linearen und nichtlinearen Dimensionsreduktionen Hauptkomponentenanalyse, Multidimensional Scaling, Isomap, Locally Linear Embedding und Diffusion Maps vermittelt. Im folgenden Kapitel wird die Rückabbildung eingeführt, welche die vorgestellten Methoden zur Dimensionsreduktion zu invertieren versucht. Um diese als Algorithmus zu formulieren, wird deren Problemstellung mit Hilfe von Lagrange-Multiplikatoren gelöst. Anschließend wird der Aufwand beziffert.

Die Rückabbildung wird in Kapitel 4 auf zwei synthetische Beispiele in Form von Polynomevaluierungen und die zwei realen Problemstellungen angewandt, um ihre Funktionalität zu analysieren. Dabei werden die Ergebnisse der verschiedenen Anwendungen

untereinander, sowie mit einer direkten Interpolation der Simulationsergebnisse ohne Verwendung der Dimensionsreduktion verglichen. Aufkommende Schwachstellen des Algorithmus werden anwendungsspezifisch erläutert und behoben, sodass sich die Approximationsergebnisse verbessern.

Im abschließenden Kapitel werden die Ergebnisse zusammengefasst und bezogen auf die möglichen Anwendungen und Weiterentwicklungen des untersuchten Algorithmus ein Ausblick gegeben.

Die graphischen Darstellungen der Einbettungen verschiedener Dimensionsreduktionen zu den Beispielen befinden sich im Anhang.

Es folgt das Abbildungsverzeichnis, in dem Parametereinstellungen ergänzend vermerkt sind, sowie das Literaturverzeichnis.

Danksagung

An dieser Stelle möchte ich meine Dankbarkeit für die Unterstützung, welche ich von allen Seiten erfahren habe, ausdrücken:

Zunächst bedanke ich mich herzlich bei Prof. Dr. Garcke für die Anregung des Themas sowie die Betreuung der Bachelorarbeit; ich hatte immer die Möglichkeit, Unklarheiten zu klären und konstruktive Kritik einzuholen. Des Weiteren möchte ich mich bei Prof. Dr. Schweitzer für die Übernahme der Zweitkorrektur und insbesondere bei Rodrigo Iza-Teran für seine jederzeit herausragende Betreuung bei Fraunhofer SCAI bedanken. Ich spreche meinen Dank auch Mandar Satyanath Pathare und Nikhil Prashanth Prabakaran für ihre geduldigen Hilfestellungen aus.

Ebenfalls gilt mein Dank Raoul, Leona und Lara, welche mir von allen möglichen Orten aus aufmerksame Anregungen zukommen ließen. In besonderem Maße gebührt mein Dank jedoch Jan für seine ausdauernde und überragende Unterstützung in der entscheidenden Endphase.

Außerdem danke ich Bruno für den starken emotionalen Rückhalt sowie meinen Eltern, welche stets ein offenes Ohr für meine Sorgen haben und auf deren Hilfe ich jederzeit zählen kann.

2. Dimensionsreduktion

Hochdimensionale Daten sind schwierig in Handhabung und Analyse, da die Komplexität von Berechnungen mit steigender Dimension exponentiell steigt und das Auffinden ausschlaggebender Unterschiede zwischen verschiedenen Datenpunkten wegen Unübersichtlichkeit aufwendig ist. Diese Phänomene werden unter dem Namen *Fluch der Dimension* zusammengefasst [LV07].

Falls Daten eine verborgene, niedrigdimensionale Struktur haben, wird versucht, diese zu finden. Man stellt sich diese Struktur häufig als eine Mannigfaltigkeit vor, von der die Daten gesampelt werden.

Definition 2.1 (Dimensionsreduktion, intrinsische Dimension und Einbettungsraum). Seien $\mathcal{M} \subset \mathbb{R}^d$ eine p -dimensionale Mannigfaltigkeit und $\{x^1, \dots, x^N\} = \mathcal{X} \subset \mathcal{M}$ die zugehörigen hochdimensionalen Daten. Sei ferner $\phi : U \subset \mathbb{R}^p \rightarrow \mathcal{M}$ die Parametrisierung der Mannigfaltigkeit. Dann versucht eine Dimensionsreduktion, die niedrigdimensionale Darstellung $f_{\text{red}}(x^i) = y^i \in \mathbb{R}^p$ der Daten zu ermitteln, wobei $\phi(y^i) = x^i$ ist. p wird intrinsische Dimension und $\mathcal{X}_{\text{red}} = \{y^1, \dots, y^N\}$ Einbettung genannt [LV07].

Vermutet man also, dass Daten $\{x^1, \dots, x^N\} = \mathcal{X} \subset \mathbb{R}^d$ näherungsweise auf einer p -dimensionalen Mannigfaltigkeit liegen, wird eine Dimensionsreduktion in den $\mathbb{R}^p$ angewandt. Falls diese gelingt, repräsentiert die niedrigdimensionale Darstellung der Daten deren wichtigste Eigenschaften und weist zudem eine geringere Komplexität auf.

Besonders bei Simulationsergebnissen liegen redundante Informationen und damit eine intrinsische Dimension $p \ll d$ vor. In diesem Kontext ist sie ein Maß für die Komplexität der dem untersuchten System zugrundeliegenden Dynamik. Beispielsweise lässt sich die Verformung der Stirnwand im Autocrashtest effektiv als Kombination der Schäden in drei verschiedenen Bereichen beschreiben. In diesem Fall ist die intrinsische Dimension mit drei beziffert. Weitere Eigenschaften, die alle Simulationsergebnisse gemeinsam haben, sind redundant, führen zu Unübersichtlichkeit und werden weggelassen.

Im Einbettungsraum können dann im Optimalfall die Einflüsse der verschiedenen Parameter auf das Simulationsergebnis abgelesen werden. Sind diese Zusammenhänge mehrdeutig, ist dies nicht zwangsläufig eine Verfehlung der Dimensionsreduktion, sondern potentiell Konsequenz problematischer Daten. Dieser Sachverhalt wird an späterer Stelle erneut aufgegriffen.

Ein besonders anschauliches Beispiel sind Fotoaufnahmen einer Büste mit variierender Helligkeit, sowie aus verschiedenen Perspektiven, welche beschrieben sind durch einen horizontalen und einen vertikalen Winkel. Die Fotoaufnahmen liegen im hochdimensionalen Raum, wobei die Dimension die Anzahl der Pixel eines Fotos ist. Eine erfolgreiche Dimensionsreduktion erkennt, dass die Fotos von dem gleichen Objekt gemacht wurden.

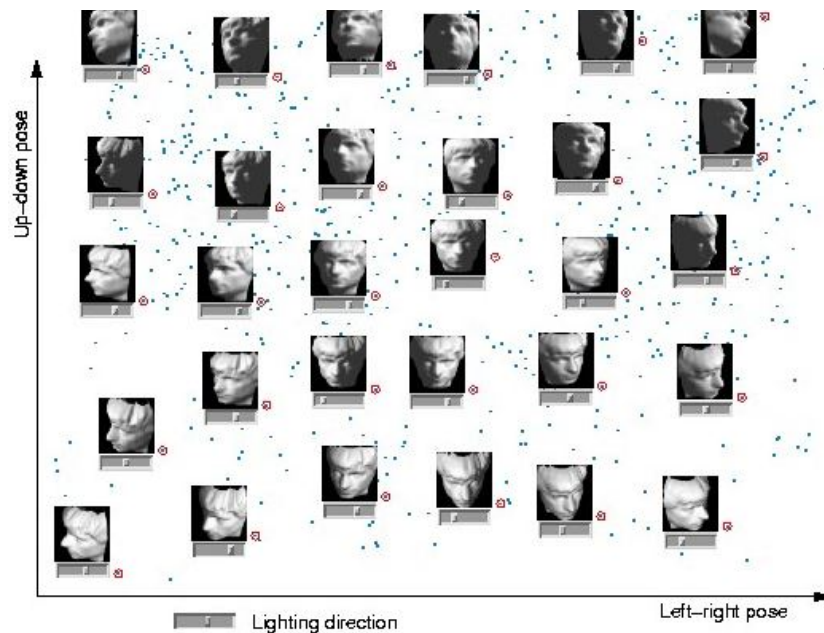


Abbildung 2.1.: Einbettung mit Isomap von 698 Aufnahmen (blaue Punkte) einer Büste in den $\mathbb{R}^3$. Isomap filtert die horizontale (x-Achse) und die vertikale (y-Achse) Blickrichtung und die Richtung des Lichteinfall (Schieberegler unter Foto) heraus. Darstellung von 30 ausgewählten Fotos, welche mit roten Kreisen gekennzeichnet sind. Quelle: [TSL00]

Die drei ausschlaggebenden Unterschiede sind lediglich die zwei verschiedenen Blickwinkel und der Lichteinfall auf die Büste. Die Projektion mit *Isomap* (siehe Kapitel 2.2.1) in einen dreidimensionalen Raum erkennt diese verborgene Struktur und sortiert die Bilder nach ihren Eigenschaften, wie in Abbildung 2.1 dargestellt.

Abschließend können folgende Voraussetzungen an eine effektive Dimensionsreduktion gestellt werden:

- Die Daten $\mathcal{X} \subset \mathbb{R}^d$ liegen näherungsweise auf einer p -dimensionalen Mannigfaltigkeit.
- Die Samples $\mathcal{X}$ wurden ausreichend dicht gewählt, sodass die entscheidenden Eigenschaften gut zum Ausdruck kommen.
- Das Rauschen der Daten ist gering.

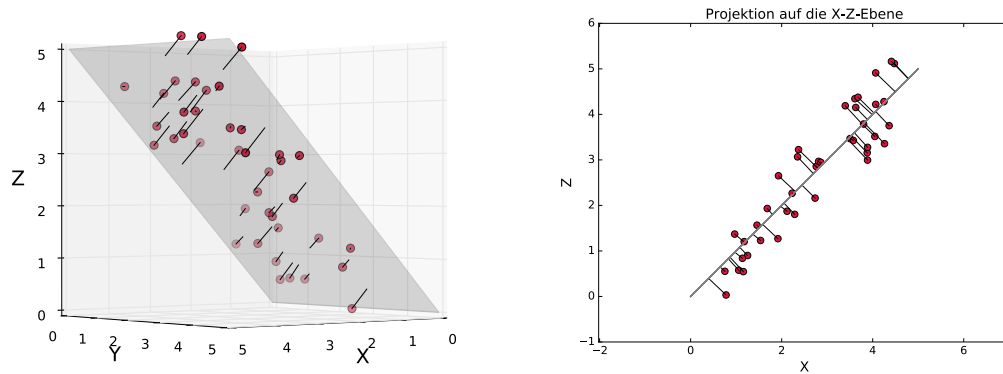


Abbildung 2.2.: Veranschaulichung einer PCA angewandt auf dreidimensionale Daten, welche die intrinsische Dimension 2 haben und daher auf eine Ebene projiziert werden

2.1. Lineare Dimensionsreduktion

Betrachte im Folgenden die Datenmatrix $X = [x^1, \dots, x^N] \in \mathbb{R}^{d \times N}$ zu den zentrierten Daten $\{x^1, \dots, x^N\} = \mathcal{X}$ und die zugehörige *Singulärwertzerlegung*.

Satz 2.1. *Sei X eine Matrix im $\mathbb{R}^{d \times N}$. Dann existiert eine Singulärwertzerlegung von X der Form*

$$X = U\Sigma V^T.$$

Dabei sind $U = [u_1, \dots, u_d] \in \mathbb{R}^{d \times d}$ und $V = [v_1, \dots, v_N] \in \mathbb{R}^{N \times N}$ unitäre Matrizen. $\Sigma \in \mathbb{R}^{d \times N}$ ist eine rechteckige Diagonalmatrix mit den eindeutigen, positiven Singulärwerten $\sigma_1 \geq \dots \geq \sigma_d$.

Die Spalten $\{u_1, \dots, u_d\} \subset \mathbb{R}^d$ der Matrix U heißen linke Singulärvektoren von X und sind orthonormal zueinander. Sie spannen den von den Datenpunkten erzeugten Raum $\text{span}(x^1, \dots, x^N)$ auf [Han09; LV07].

Der Betrag des Singulärwerts gibt dabei die Wichtigkeit des zugehörigen Singulärvektors an, denn ein großer Singulärwert bedeutet, dass der Vektor bei den Linearkombinationen der Datenpunkte großes Gewicht hat. Der Spann der p linken Singulärvektoren zu den p betragsgrößten Singulärwerten identifiziert also die im obigen Sinne charakterisierende Struktur der Datenmenge $\mathcal{X}$ in p Dimensionen, weshalb diese Vektoren Hauptkomponenten heißen [Jol86].

Es lässt sich das Verfahren *Hauptachsentransformation* (Principal Component Analysis, PCA, Abbildung 2.2) definieren, welches die Daten orthogonal auf diese p -dimensionale Hyperebene projiziert.

Satz 2.2. *Seien $X = [x^1, \dots, x^N] \in \mathbb{R}^{d \times N}$ die Datenmatrix zu N zentrierten Datenpunkten im $\mathbb{R}^d$, $X = U\Sigma V^T$ die zugehörige Singulärwertzerlegung und $p < d$. Für das*

Rekonstruktionsproblem

$$\min_{P \in \mathbb{R}^{d \times p}} \sum_{i=1}^N \|x^i - PP^T x^i\|_2^2 \quad \text{u.d.N.} \quad P^T P = I_p$$

ist die Lösung gegeben durch $U_p = [u_1, \dots, u_p]$. Die eingebetteten Datenpunkte $\{y^1, \dots, y^N\}$ erhält man mittels

$$Y = [y^1, \dots, y^N] = I_{p \times d} \Sigma V^T = U_p^T X.$$

Mit dem Term

$$\sum_{i=p+1}^d \sigma_i^2$$

lässt sich der Fehler des obigen Verfahrens abschätzen [LV07; Laf04].

Der Fehlerabschätzung wegen bestimmt die Anzahl der Singulärwerte, welche eindeutig größer null sind, die intrinsische Dimension p .

Es lässt sich zeigen, dass obiges Verfahren äquivalent ist zur Errechnung einer Einbettung $y^1, \dots, y^N \in \mathbb{R}^p$, sodass

$$\sum_{i,j=1}^N |d_E^2(x^i, x^j) - d_E^2(y^i, y^j)|$$

minimal ist, wobei $d_E(x, y) = \|x - y\|_2$ der euklidische Abstand ist. Diese Einbettung ist also gleichzeitig jene, die die euklidischen Abstände zwischen den Datenpunkten am besten erhält, weshalb der Algorithmus auch *Classical Multidimensional Scaling* (CMDS) genannt wird.

Mit $H = I - \frac{1}{N} \mathbb{1} \mathbb{1}^T$ der Zentrierungsmatrix und D der euklidischen Distanzmatrix zu $\mathcal{X}$ gilt

$$\begin{aligned} -\frac{1}{2} H D H &= X^T X \\ &= (U \Sigma V^T)^T (U \Sigma V^T) \\ &= V \Sigma^2 V^T. \end{aligned}$$

Somit stimmen die Eigenwerte von $-\frac{1}{2} H D H$ mit den quadrierten Singulärwerten von X überein [LV07]. Daraus ergibt sich der Algorithmus des CMDS bzw. der PCA (siehe Algorithmus 1).

Natürlich kann die Einbettung auch durch eine Singulärwertzerlegung von X oder eine Eigenwertzerlegung von $X^T X$ berechnet werden. In dieser Arbeit wurde aber die Darstellung mit $-\frac{1}{2} H D H$ gewählt, da die Dimensionsreduktion in den folgenden Kapiteln mittels Modifikationen an der Distanzmatrix angepasst wird. Somit kann der Algorithmus des CMDS als Grundlage der weiteren Methoden verwendet werden, indem lediglich eine andere Matrix D eingegeben wird.

Algorithmus 1 : CMDS

Data : Euklidische Distanzmatrix D , intrinsische Dimension p

Result : Einbettung Y in p Dimensionen

begin

$$\left[\begin{array}{l} G = -\frac{1}{2}HDH \\ [V_p, \Lambda_p] = \text{EVD}(G, p) \end{array} \right.$$

return $\Lambda_p^{\frac{1}{2}} V^T$

2.2. Nichtlineare Dimensionsreduktion

Liegen die Daten auf einer nichtlinearen Mannigfaltigkeit, funktionieren eine PCA, bzw. das CMDS, nur bedingt gut. Das veranschaulicht die Swissroll, da sie im zweidimensionalen Fall eine Spirale, damit einer Linie topologisch äquivalent ist und eine eindimensionale Mannigfaltigkeit darstellt.

Jedoch ist der Versuch eine eindimensionale Hyperebene über die Daten zu legen, sodass diese gut approximiert werden, vergeblich. Eine PCA findet die zugrundeliegende eindimensionale Mannigfaltigkeit also nicht. Auch mit dem Ansatz des CMDS ist die Ursache des Misserfolges schnell gefunden; die euklidischen Abstände, welche beim CMDS eingebettet werden, stimmen nicht immer mit den eigentlichen Abständen der Punkte auf der Mannigfaltigkeit überein. Dieses Problem wird in Abbildung 2.3a deutlich.

2.2.1. Isomap

Im Rahmen einer zielführenden Dimensionsreduktion werden die euklidischen Distanzen beim CMDS durch die Abstände auf der Mannigfaltigkeit, welche *geodätische Abstände* genannt werden, ersetzt. Da die zugrundeliegende Mannigfaltigkeit nicht bekannt ist, werden die geodätischen Abstände abgeschätzt, wofür im Falle von Isomap ein *Nachbarschaftsgraph* benötigt wird:

Definition 2.2 (*k*-nächste-Nachbarn-Graph). Sei $\mathcal{X} \subset \mathbb{R}^d$ eine Datenmenge und $\mathcal{J}_{\mathcal{X}} : \mathcal{X} \rightarrow \{1, \dots, N\}^k$ ordne jedem $x \in \mathcal{X}$ die Indizes seiner k nächsten Nachbarn in $\mathcal{X}$ zu. Dann ist der zugehörige *k*-nächste-Nachbarn-Graph der Graph $G(V, E)$ mit der Knotenmenge $V = \mathcal{X}$. Die Kantenmenge $E \subset \mathcal{X} \times \mathcal{X}$ werde so gewählt, dass

$$\{x^i, x^j\} \in E \Leftrightarrow j \in \mathcal{J}_{\mathcal{X}}(x^i)$$

[LV07].

In dem Graphen sind zwei Knoten also genau dann verbunden, wenn die zugehörigen Elemente in $\mathcal{X}$ nahe sind, vergleiche dazu Abbildung 2.4a. Die geodätischen Abstände werden im Folgenden mit Hilfe der *Graphenabstände* abgeschätzt:

Definition 2.3 (Graphendistanz). Sei $G(V, E)$ der zu $\mathcal{X}$ gehörige *k*-nächste-Nachbarn-Graph (*k*-nearest-neighbors-graph) und $P_{x,z}^* \subseteq E$ der kürzeste Pfad von x nach z . Dann

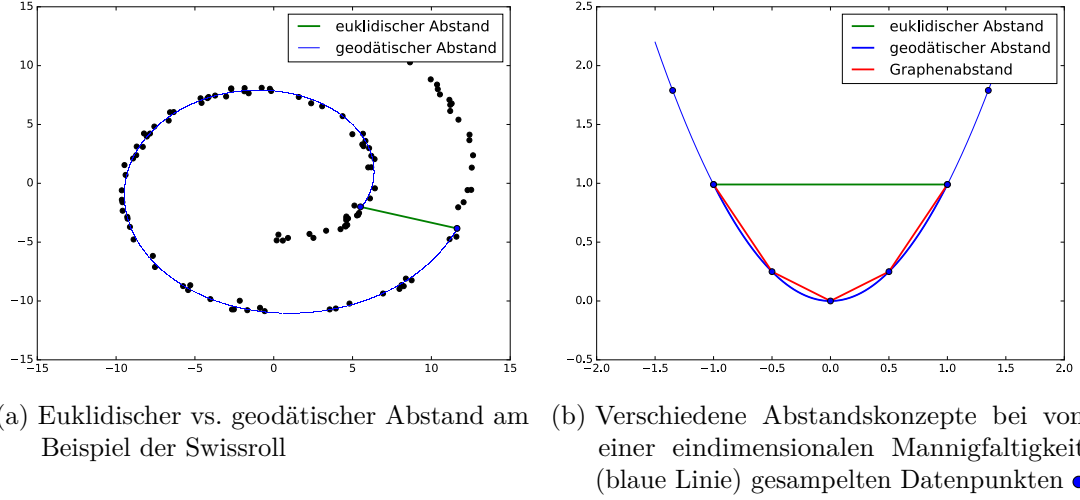


Abbildung 2.3.: Abstandskonzepte bei Dimensionsreduktionen

ist die Graphendistanz von $x, z \in \mathcal{X}$ definiert als

$$d_G(x, z) = \sum_{\{v, w\} \in P_{x, z}^*} d_E(v, w),$$

wobei d_E der euklidische Abstand ist [TSL00].

Der geodätische Abstand benachbarter Datenpunkte wird also letztendlich durch ihre euklidische Distanz approximiert. Die verschiedenen Konzepte werden in Abbildung 2.3b graphisch dargestellt.

Schlussendlich ergibt sich für die Dimensionsreduktion Isomap folgende Vorgehensweise:

1. Konstruktion des k -nächste-Nachbarn-Graph
2. Bestimmung der kürzesten Pfade zwischen allen Datenpunkten in $\mathcal{X}$, zum Beispiel mit den Algorithmen von Dijkstra oder Floyd-Warshall [KV12]
3. Berechnung der Graphendistanzen $d_G(x^i, x^j)$ für alle $x^i, x^j \in \mathcal{X}$
4. Anwendung des Multidimensional Scaling auf die Graphendistanzen

$$\min_{\{y^1, \dots, y^N\} \subset \mathbb{R}^p} \sum_{i, j=1}^N |d_G^2(x^i, x^j) - d_E^2(y^i, y^j)|$$

Ersetzen der euklidische Distanzmatrix aus Algorithmus 1 durch die Graphendistanzmatrix

$$(D_G)_{i, j} = d_G(x^i, x^j)$$

liefert eine kompakte Darstellung des Verfahrens wie folgt:

Algorithmus 2 : Isomap

Data : Datenmatrix X , Nachbarschaftsgröße k , intrinsische Dimension p

Result : Einbettung Y in p Dimensionen

begin

 Erstelle den Nachbarschaftsgraphen $G(V, E)$ mit Nachbarschaftsgröße k

$D_G = [0]_{i,j=0}^N$

forall i, j **in** $\{1, \dots, N\}$ **do**

$D_G[i][j] = d_G(x^i, x^j)$

$G = -\frac{1}{2}HD_GH$

$[V_p, \Lambda_p] = \text{EVD}(G, p)$

return $\Lambda_p^{\frac{1}{2}}V^T$

Die Graphendistanzen approximieren die geodätischen Distanzen auf der Mannigfaltigkeit mit hoher Güte, sofern die Datenpunkte in Abhängigkeit der Topologie der Mannigfaltigkeit dicht genug gesampelt sind, um jegliche Eigenschaften derselben ausreichend darzustellen [TSL00]. Außerdem muss im Falle starken Rauschens die Anzahl der Samples erhöht werden, um „Abkürzungen“ vorzubeugen.

Das bedeutet bei der Swissroll, dass ein Knoten einem Knoten auf einer anderen Windung der Schnecke nie näher sein darf, als den eigentlichen Nachbarknoten auf der Mannigfaltigkeit. Dies wird in Abbildung 2.4b ersichtlich und ist ein Zeichen unzureichenden Samplings oder heftiger Interferenzen.

Da es schwierig ist, die Qualität einer Einbettung mathematisch zu bewerten, wird die Dimensionsreduktion mit Isomap für verschiedene Nachbarschaftsgrößen durchgeführt und für jede Anwendung das beste Ergebnis manuell ausgewählt.

Erste Anwendung Isomaps

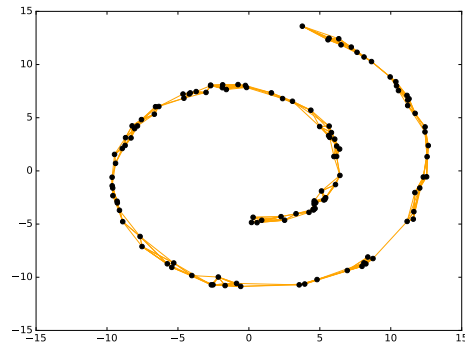
Nun soll die Funktionsweise von Isomap am Beispiel von Polynomen zweiten Grades verdeutlicht werden. Man betrachte Polynome der Form $P_{p_1, p_2}(x) = p_1x + p_2x^2$ mit 40 willkürlich gewählten Koeffizientenpaaren $\{p^i = (p_1^i, p_2^i)\} = \Omega \subset [-0.4, 0.4] \times [-0.1, 0.1]$, $i = 1, \dots, 40$ und die Abbildung

$$f_{sim} : \mathcal{P} \rightarrow \mathbb{R}^{1000}$$
$$f_{sim}(p_1, p_2) = \begin{pmatrix} P_{p_1, p_2}(z_1) \\ \vdots \\ P_{p_1, p_2}(z_{1000}) \end{pmatrix}, \quad (2.1)$$

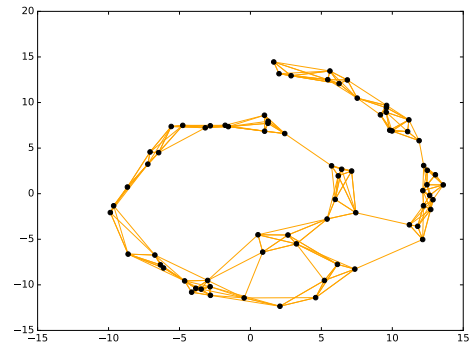
$z_j : 1000$ äquidistante Stützstellen in $[-3, 3]$

mit $d = 1000$, welche als Simulation interpretiert wird (siehe Abbildung 2.5). Ferner seien $x^i = f_{sim}(p_1^i, p_2^i)$ für $i = 1, \dots, 40$.

Generiert werden also vierzig verschiedene Polynome und an je 1000 Stützstellen evaluiert. Es sei $\{x^1, \dots, x^{40}\} = \mathcal{X} \subset \mathbb{R}^{1000}$ der Simulationsraum, welcher also die Graphen der 40 Polynome enthält.

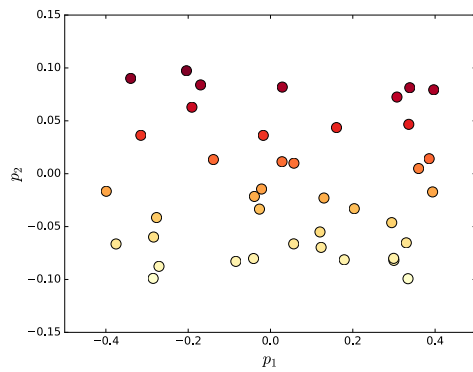


(a) Nachbarschaftsgraph ohne Abkürzungen

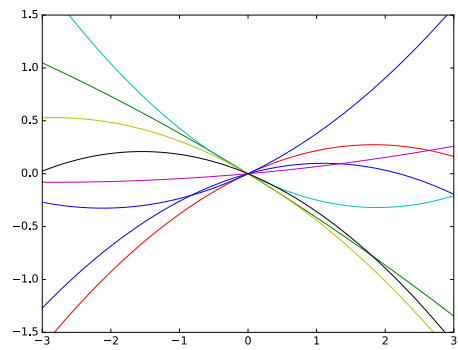


(b) Graph mit Abkürzungen aufgrund starken Rauschens

Abbildung 2.4.: Nachbarbarschaftsgraphen zwei verschiedener Swissrolls. Nachbarschaftsgröße $k=5$



(a) Parameterraum $\mathcal{P}$



(b) Menge der Polynomevaluationen $\mathcal{X}$ (Simulationsraum)

Abbildung 2.5.: Synthetisches Beispiel POLY2

	Nachbarn in P		Nachbarn in X_{red}	
	<i>Index</i>	<i>Distanz in $\mathcal{X}$</i>	<i>Index</i>	<i>Distanz in $\mathcal{X}$</i>
1.	13:	0.104717	13:	0.104717
2.	38:	0.224201	38:	0.224201
3.	23:	1.168844	19:	0.530807
4.	4:	1.517936	36:	0.684404
5.	19:	0.530807	15:	0.752132
6.	27:	2.032097	9:	1.460899

Tabelle 2.1.: Nächsten Nachbarn eines Samples im Parameterraum P und im Einbettungsraum $\mathcal{X}_{red}$ mit ihren Distanzen im Simulationsraum. Die nächsten Nachbarn aus dem Einbettungsraum in $\mathcal{X}$ liegen näher an dem Sample als die Nachbarn aus dem Parameterraum.

Die Betrachtung der Eigenwerte der Distanzmatrix offenbart eine intrinsische Dimension $p = 2$. Wenn nun auf diese hochdimensionale Menge zur Dimensionsreduktion auf Dimension 2 Isomap mit Nachbarschaftsgröße 7 angewendet wird, erhält man den zweidimensionalen Einbettungsraum $\mathcal{X}_{red}$. Es lässt sich sehr gut erkennen, dass der Algorithmus den Einfluss der beiden Koeffizienten tatsächlich bestätigt (vgl. Abbildung A.2).

Unter genauer Betrachtung zeigt sich, dass die von Isomap gefundene Einbettung die Nachbarschaftsverhältnisse in $\mathcal{X}$ sogar besser darstellt. Untersuche dafür die 5 nächsten Nachbarn eines Punktes $p^j \in \mathcal{P}$, $j \in \{1, \dots, N\}$ und des korrespondierenden Punktes $y^j \in \mathcal{X}_{red}$, um die zugehörigen Abstände der Datenpunkte im Simulationsraum $\mathcal{X}$ (siehe Tabelle 2.1) zu vergleichen. Die nächsten Nachbarn aus der Einbettung $\mathcal{X}_{red}$ liegen im Simulationsraum näher an x^j in $\mathcal{X}$, als ihre Gegenstücke aus dem Parameterraum.

Das Ziel Isomaps, die Nachbarschaftsbeziehungen der Simulationen in $\mathcal{X}$ sinnvoll zu repräsentieren, wird also in diesem Beispiel vollständig erreicht.

2.2.2. Locally Linear Embedding

Isomap beachtet bei der Dimensionsreduktion auch Beziehungen weit voneinander entfernter Datenpunkte, indem das Verfahren sie gemäß ihrer Graphenabstände, welche die geodätischen Abstände auf der Mannigfaltigkeit approximieren sollen, einbettet.

Das *Locally Linear Embedding* (LLE) versucht ebenfalls, eine nichtlineare Dimensionsreduktion durchzuführen, bezieht sich dabei jedoch nicht auf Abstände einander ferner Samples und nutzt stattdessen lokale Symmetrien. Es bettet unter Durchführung folgender drei Schritte exklusiv lokale, lineare Informationen in Gestalt euklidischer Abstände zu den k nächsten Nachbarn ein, siehe Abbildung 2.6 [RS00; LV07].

1. Ermittlung der k nächsten Nachbarn von x^i mit den Indizes $\mathcal{J}_X(x^i)$ für $i = 1, \dots, N$

2. Darstellung der Datenpunkte x^i als Linearkombinationen ihrer k nächsten Nachbarn

$$\min_{W \in \mathbb{R}^{N \times N}} \sum_{i=1}^N \left\| x^i - \sum_{j \in \mathcal{J}_X(x^i)} W_{i,j} x^j \right\|_2^2 \quad \text{u.d.N.} \quad \sum_{j=1}^N W_{i,j} = 1 \quad (2.2)$$

3. Suche einer Einbettung $\mathcal{X}_{red} = \{y^1, \dots, y^N\} \subset \mathbb{R}^p$, sodass die optimalen Linearkombinationen mit den Gewichten W^* , welche (2.2) minimieren, bestmöglich eingebettet werden.

$$\min_{\{y^1, \dots, y^N\} \subset \mathbb{R}^p} \sum_{i=1}^N \left\| y^i - \sum_{j=1}^N W_{i,j}^* y^j \right\|_2^2$$

Das LLE nutzt zur Einbettung der Linearkombinationen die Eigenschaft, dass diese invariant sind unter linearen Abbildungen. Indem der Nebenbedingung $\sum_{j=1}^N W_{i,j} = 1$ Geltung verschafft wird, werden große Gewichte unterbunden. Unter dieser Voraussetzung sind obige Linearkombinationen zusätzlich translationsinvariant; den Beweis folgend als Lemma:

Lemma 2.1. *Seien $\omega_1, \dots, \omega_N \in \mathbb{R}$ und $v_1, \dots, v_N, t \in \mathbb{R}^d$. Dann gilt:*

$$\sum_{j=1}^N \omega_j = 1 \quad \Rightarrow \quad \sum_{j=1}^N \omega_j v_j + t = \sum_{j=1}^N \omega_j (v_j + t)$$

Beweis.

$$\begin{aligned} \sum_{j=1}^N \omega_j v_j + t & \stackrel{(\sum_j \omega_j = 1)}{=} \sum_{j=1}^N \omega_j v_j + \left(\sum_{j=1}^N \omega_j \right) t \\ & = \sum_{j=1}^N \omega_j v_j + \sum_{j=1}^N (\omega_j t) \\ & = \sum_{j=1}^N \omega_j (v_j + t) \end{aligned}$$

□

Vermöge der Auffassung, dass die Datenpunkte lediglich aus den k nächsten Nachbarn rekonstruiert werden, zählt das LLE zu den nichtlinearen Einbettungen, obwohl ausnahmslos lineare Zusammenhänge der hochdimensionalen Daten erfasst werden [RS00].

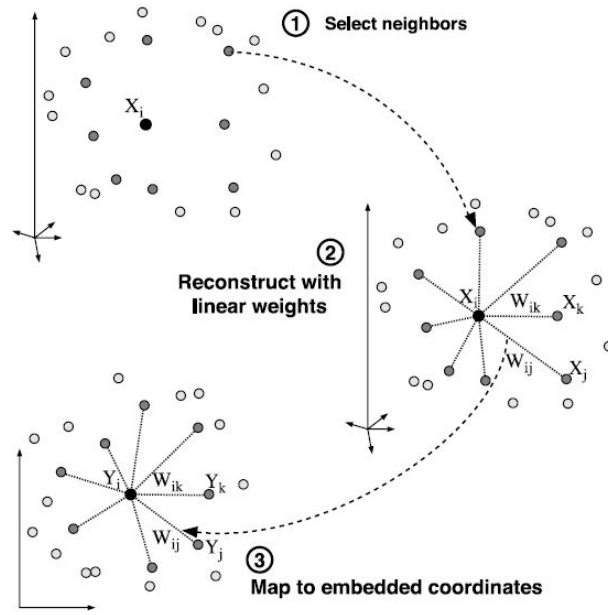


Abbildung 2.6.: Graphische Darstellung der einzelnen Schritte des LLE. Quelle: [RS00]

2.2.3. Diffusion Maps

Diffusion Maps ist ein weiterer Algorithmus zur nichtlinearen Dimensionsreduktion, der mit Hilfe einer Diffusionsmetrik die geodätischen Abstände auf Mannigfaltigkeiten beschreibt. Der Abstand zwischen u und z in $\mathcal{X}$ unter dieser Metrik ist der Diffusion in u eingeschleufter Wärme bis zum Datenpunkt z in q Zeitschritten nachempfunden [NLK06; CL06].

Um mittels Multidimensional Scalings eine Einbettung finden zu können, wird eine nach der Diffusionsmetrik erstellte Distanzmatrix benötigt. Deren Konstruktion gelingt in folgenden Schritten [CL06; Gri+14]:

1. *Initialisierung eines Nachbarschaftsgraphen* $G(V, E)$ mit $V = \mathcal{X}$ und $E = \mathcal{X} \times \mathcal{X}$
Die Kantengewichte des vollständigen Graphen sind gegeben durch eine Funktion

$$w_e(u, z) = \exp\left(\frac{-d^2(u, z)}{\epsilon}\right)$$

für alle $u, z \in \mathcal{X}$, d ein Abstandsmaß. In dieser Arbeit wird in Verbindung mit Diffusion Maps die euklidische Distanz d_E verwendet, sodass w_e dem vertrauten Gauss-Kern entspricht.

2. *Definition einer Markov-Kette auf dem Graphen*

Definiere die Übergangswahrscheinlichkeiten der Markov-Kette als

$$p(u, z) = \frac{w_e(u, z)}{\sum_{v \in \mathcal{X}} w_e(u, v)},$$

sodass diese ansteigen, wenn zwei Datenpunkte nahe beieinander sind.

3. *Berechnung der Diffusionsabstände, durch q -malige Iteration der Markov Kette*

$$d_{Diff,q}(u, z) = \sum_{v \in \mathcal{X}} \frac{(p_q(u, v) - p_q(v, z))^2}{\phi_0(v)}$$

Dabei beschreibt $p_q(u, z)$ die Übergangswahrscheinlichkeit von u nach z in q Zeitschritten, und $\phi_0(v)$ die stationäre Verteilung. Dementsprechend nimmt $d_{Diff,q}(u, z)$ einen kleinen Wert an, falls zwischen u und z viele Pfade der Länge $2 \cdot q$ existieren, die Punkte folglich ähnlich sind.

4. *Einbettung der Diffusionsabstände*

Multidimensional Scaling bettet die Datenpunkte unter Berücksichtigung ihrer Diffusionsabstände $d_{Diff,q}(u, z)$ in den $\mathbb{R}^p$ ein, sodass diese effektiv in euklidische Abstände überführt werden. Dies ist äquivalent zu der Operation

$$\Psi : \mathcal{X} \rightarrow \mathbb{R}^p$$

$$v \mapsto \begin{pmatrix} \lambda_1^q \psi_1(v) \\ \lambda_2^q \psi_2(v) \\ \vdots \\ \lambda_p^q \psi_p(v) \end{pmatrix},$$

wobei λ_j und ψ_j den Eigenwerten und rechten Eigenvektoren der Wahrscheinlichkeitsmatrix P mit $P_{i,j} = p(x^i, x^j)$ entsprechend. Dies hat den Vorteil, dass die Diffusionsabstände nicht explizit berechnet werden müssen [Laf04].

In dem Sinne, dass durch die Berechnung aller Übergangswahrscheinlichkeiten auch die Beziehung weit voneinander entfernter Punkte zum Ausdruck kommt, agiert Diffusion Maps weniger lokal als Isomap. Ebenso ist der Algorithmus stabiler bei verrauschten Datensätzen, da bei der Ermittlung der Wahrscheinlichkeiten über alle möglichen Pfade auf dem Graph summiert wird [Laf04]. Jedoch ordnet diese Methode zwei von einem Engpass getrennten Datenpunkten eine weite Distanz zu, da der enge Abschnitt nur wenige Wege passieren lässt (Abbildung 2.7). Dies ist indes a priori nicht angebracht.

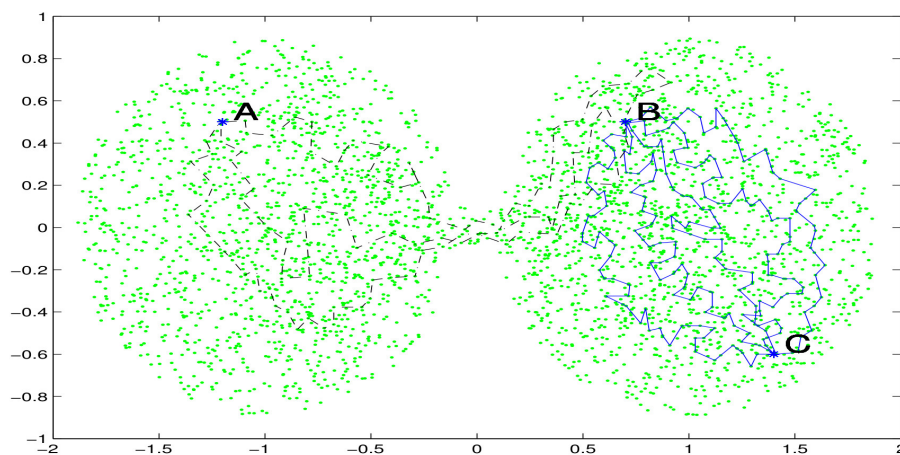


Abbildung 2.7.: In der Diffusionsmetrik sind die Punkte B und C nahe beieinander, da sie von vielen Wege auf dem Nachbarbarschaftsgraphen verknüpft sind. Im Gegensatz dazu werden B und A wegen des Engpasses in der Mitte des Datensatzes als weit auseinander eingestuft, weil nur wenige verbindende Pfade existieren. Quelle: [Laf04]

3. Die Rückabbildung

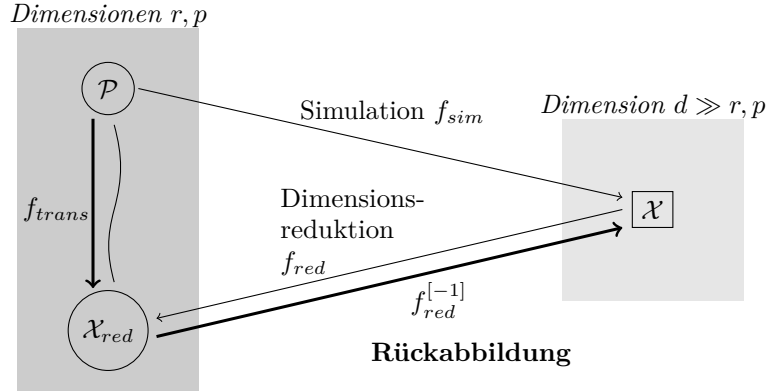


Abbildung 3.1.: Übersicht an der Rückabbildung beteiligter Funktionen und Räume

Die Rückabbildung approximiert die Simulation $f_{sim} : \mathcal{P} \rightarrow \mathbb{R}^d$ für ein $p^* \in \mathcal{P} \setminus \Omega$ unter Verwendung bereits vorliegender Simulationsergebnisse $\mathcal{X} \subset \mathbb{R}^d$ zu Parametereinstellungen $\Omega \subset \mathcal{P} \subseteq \mathbb{R}^r$ und einer Einbettung $\mathcal{X}_{red} \subset \mathbb{R}^p$.

Ohne Kenntnis der intrinsischen Struktur der Simulation f_{sim} bietet sich eine direkte Interpolation in $\mathcal{X}$ zu dessen Approximation nicht an; der Einfluss von Parametervariationen auf das Ergebnis von f_{sim} ist in physikalischen Gesetzmäßigkeiten der Systemdynamik begründet, hochkomplex und nicht unmittelbar auszunutzen.

Darüber hinaus interpolieren naheliegende Verfahren, beispielsweise die kernbasierten Interpolationen, nur in eindimensionalen Räumen, sodass diese Methoden in $\mathcal{X}$ zwangsläufig koordinatenweise angewandt würden. Ist $p < r$, der Dimension von $\mathcal{P}$, liegt Redundanz vor, sodass eine in $\mathcal{X}$ durchgeführte Interpolation unverhältnismäßig rechenintensiv wird. Dies verdeutlicht an späterer Stelle die Anwendung ZUGVERSUCH.

Insgesamt stellt eine Interpolation in einem hochdimensionalen Raum in vielerlei Hinsicht eine außerordentliche Herausforderung dar. Damit ist die Rückabbildung als alternatives Verfahren von besonderem Interesse.

3.1. Funktionsweise

Betrachte an dieser Stelle erneut Tabelle 2.1, welche veranschaulicht, dass, unter Voraussetzung einer guten Einbettung mit Isomap, die Darstellung der Nachbarschaftsbeziehungen in $\mathcal{X}$ durch die nächsten Nachbarn in $\mathcal{X}_{red}$ besser gelingt, als durch die in $\mathcal{P}$.

Die Rückabbildung macht sich solche Informationen aus $\mathcal{X}_{red}$ bei der Vorhersage des Simulationsergebnisses $f_{sim}(p^*)$ zu einem $p^* \in \mathcal{P}$ zu eigen.

Mit gegebenen $p^*, \Omega \subset \mathcal{P}, \mathcal{X}$ und $\mathcal{X}_{red}$ sind zu diesem Zweck zwei Schritte notwendig [RS00; Fra15]:

1. Interpoliere in $\mathcal{X}_{red}$ unter Verwendung der Stützstellen in Ω und approximiere die Einbettung des Simulationsergebnisses $\hat{y} = f_{red} \circ f_{sim}(p^*)$.
2. Nutze aus $\mathcal{X}_{red}$ gewonnene Kenntnis der Nachbarschaftsbeziehungen in $\mathcal{X}$, um die Dimensionsreduktion zu invertieren

Die Interpolation in Punkt 1 ist möglich, angenommen eine Strukturanalogie zwischen $\mathcal{P}$ und $\mathcal{X}_{red}$ liege vor. In [Fra15] wird hierfür eine Radiale-Basisfunktion-Interpolation mit dem thin-plate-spline als radialen Kern verwendet; anwendungsbezogen kann die Art der Approximation bei Bedarf variiert werden. Im Folgenden werde diese Interpolation mit f_{trans} bezeichnet, um auf ihre Wirkung als geometrische Transformation hinzuweisen.

Das eigentliche Herzstück des Algorithmus ist dementsprechend $f_{red}^{[-1]}$, die Inversion von Isomap in Punkt 2. In diesem zweiten Teil der Rückabbildung wird eine *lokale lineare Interpolation* mit der Abkürzung LLI verwendet.

Nach erfolgter Einbettung mit Isomap entsprechen die euklidischen Abstände zweier Punkte in $\mathcal{X}_{red}$ den Graphenabständen in $\mathcal{X}$, welche wiederum die geodätischen Abstände approximieren. Außerdem korrespondiert der Graphenabstand zweier Punkte in $\mathcal{X}$, sofern benachbart, mit deren euklidischem Abstand. Dementsprechend stimmen die euklidischen Abstände in $\mathcal{X}$ und $\mathcal{X}_{red}$ zwischen nahe gelegenen Samples ungefähr überein.

Einen Punkt $y^i \in \mathcal{X}_{red}, i \in \{1, \dots, N\}$ als Linearkombination seiner nächsten Nachbarn $\{y^j | j \in \mathcal{J}\}$ repräsentiert, und deren Gewichte auf eine Linearkombination der Punkte $\{x^j | j \in \mathcal{J}\}$ in $\mathcal{X}$ übertragen, liefert aus diesem Grund eine Annäherung seines Gegenstückes $x^i \in \mathcal{X}$. Dabei bezeichne $\mathcal{J}$ wie in Definition 2.2 die Menge der Indizes der k nächsten Nachbarn in $\mathcal{X}_{red}$.

Ebendieser Ansatz bringt $f_{red}^{[-1]}$ hervor; explizit wird die Ausgabe von $f_{trans}(p^*) = y^* \approx \hat{y} \in \mathcal{X}_{red}$ als Linearkombination der nächsten Nachbarn $\{y^j | j \in \mathcal{J}\}$ in $\mathcal{X}_{red}$ dargestellt

$$y^* = f_{trans}(p^*) \approx \sum_{j \in \mathcal{J}} \omega_j^* y^j.$$

Die Gewichte $\omega^* \in \mathbb{R}^k$ werden in einer Linearkombination der dazugehörigen Punkte $\{x^j | j \in \mathcal{J}\} \subset \mathcal{X}$ verwendet, und schlussendlich

$$f_{red}^{[-1]} \circ f_{trans}(p^*) = \sum_{j \in \mathcal{J}} \omega_j^* x^j \approx f_{sim}(p^*).$$

Diese Vorgehensweise ist dem LLE entnommen (Kapitel 2.2.2), bei welcher dieser Ansatz der Rekonstruktion lokaler Linearkombinationen verwendet wurde, um hochdimensionale Daten einzubetten. Es bleibt, die Gewichte der ursprünglichen Linearkombination in $\mathcal{X}_{red}$ zu ermitteln. Auch an dieser Stelle lehnen wir uns an die im Rahmen des

LLE demonstrierte Methodik an (2.2); die Fragestellung wird als Optimierungsproblem, allerdings mit zusätzlichem Strafterm, formuliert.

Minimiere also über alle Gewichtsvektoren $\omega = (\omega_j)_{j \in \mathcal{J}} \in \mathbb{R}^k$ die folgende Bedingung, um die gesuchte Linearkombination in $\mathcal{X}_{red}$ zu finden:

$$\min_{\omega \in \mathbb{R}^k} \left\| y^* - \sum_{j \in \mathcal{J}} \omega_j y^j \right\|_2^2. \quad (3.1)$$

Analog zum LLE im Kapitel 2.2.2 wird zu (3.1) eine Normalisierung hinzugefügt, sodass erstens hohe Gewichte ω_j bestraft und zweitens die Linearkombinationen translationsinvariant werden [RS00]. Da Isomap Translationen beim Zentrieren der Daten realisiert, ist jene Bedingung an die Linearkombinationen unerlässlich:

$$\min_{\omega \in \mathbb{R}^k} \left\| y^* - \sum_{j \in \mathcal{J}} \omega_j y^j \right\|_2^2 \quad \text{s.d.} \quad \sum_{j \in \mathcal{J}} \omega_j = 1. \quad (3.2)$$

Im Allgemeinen ist die Anzahl k der einbezogenen nächsten Nachbarn größer als $d + 1$, sodass die Gleichung (3.2) nicht eindeutig lösbar sein könnte. Verwende daher als stetigen Regularisierungsoperator die folgende gewichtete Norm $s : \mathbb{R}^k \rightarrow \mathbb{R}$:

$$s(\omega) := \|\omega\|_c^2 := \sum_{j \in \mathcal{J}} c_j \omega_j^2, \quad c_j = \epsilon \left(\frac{\|y^* - y^j\|_2}{\max_i \{\|y^* - y^i\|_2\}} \right)^l, \quad 0 < \epsilon \ll 1, 1 \leq l \in \mathbb{N}$$

Die Summation der einzelnen quadrierten Gewichte ω_j bestraft betragsgroße ω_j , damit alle Nachbarn bei dem Finden der Linearkombination möglichst gleichermaßen beachtet werden. Überdies fällt die Bestrafung wegen der c_j zusätzlich stark aus, wenn ein verhältnismäßig weit entfernten Nachbar y^j großen Einfluss beim Rekonstruieren des y^* hat. Die Potenz l führt dazu, dass, falls $l > 1$, die Bestrafung weitentlegener Nachbarn nicht linear mit der Distanz steigt, sondern erst ab einem gewissen Abstand ins Gewicht fällt. Der Einfluss der Potenz l auf die Straftermgewichte c_j ist in der Abbildung 3.2 ausgeführt.

Insgesamt wird so Fehlern durch einzelne Ausreißer in den Daten vorgebeugt.

Zusätzlich steuert der Hyperparameter $\epsilon < 0$ die Auswirkung der Regularisierung auf die Lösung des Optimierungsproblems.

Das Optimierungsproblem (3.2) wird somit zu

$$\min_{\omega \in \mathbb{R}^k} \left\| y^* - \sum_{j \in \mathcal{J}} \omega_j y^j \right\|_2^2 + \|\omega\|_c^2 \quad \text{u.d.N.} \quad \sum_{j \in \mathcal{J}} \omega_j = 1. \quad (3.3)$$

[Fra15; Fra+14].

Der Gewichtsvektors $w^* \in \mathbb{R}^k$, welcher (3.3) minimiert, dient unmittelbar zur Approximation von $f_{sim}(p^*)$ im hochdimensionalen Raum.

$$x^* = \sum_{j \in \mathcal{J}} \omega_j^* x^j \quad (3.4)$$

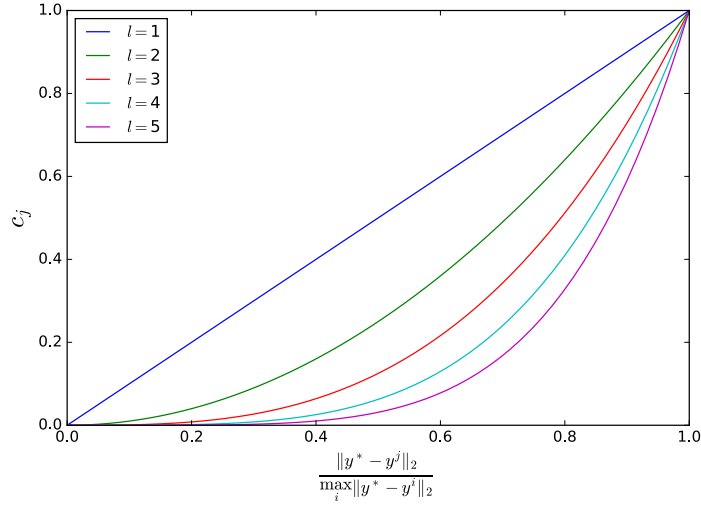


Abbildung 3.2.: Einfluss der Wahl von l auf das Straftermgewicht c_j in Abhängigkeit der normierten Entfernung des Nachbarn y^j von y^*

Zusammenfassend ergeben sich zur Durchführung der Rückabbildung:

1.	Projiziere p^* in den Einbettungsraum.	$y^* = f_{trans}(p^*)$
2.a	Finde Indices $\mathcal{J}$ der k nächsten Nachbarn von y^* .	$\{y^j \mid j \in \mathcal{J}\} \subset \mathcal{X}_{red}$
2.b	Berechne Gewichtsvektor $\omega^* \in \mathbb{R}^k$ zur Approximation von y^* im Einbettungsraum.	$\min_{\omega \in \mathbb{R}^k} \ y^* - \sum_{j \in \mathcal{J}} \omega_j y^j\ _2^2 + \ \omega\ _c^2$ $\text{u.d.N.} \quad \sum_{j \in \mathcal{J}} \omega_j = 1$ mit der gewichteten Norm $\ \omega\ _c^2 := \sum_{j \in \mathcal{J}} c_j \omega_j^2,$ $c_j = \epsilon \left(\frac{\ y^* - y^j\ _2}{\max_i \ y^* - y^i\ _2} \right)^l$ für $0 < \epsilon \ll 1$ und $1 \leq l \in \mathbb{N}$
2.c	Benutze Gewichtsvektor zur Interpolation im Simulationsraum.	$x^* = \sum_{j \in \mathcal{J}} \omega_j^* x^j$

Tabelle 3.2.: Rückabbildung in vier Schritten

Während in [Fra15] und [Fra+14] die Rückabbildung nur auf Einbettungen mit Iso-map angewandt wird, verwendet diese Arbeit sie auch in Verbindung mit Diffusion Maps. Diffusion Maps selbst bettet für nahe gelegene Nachbarn zwar keine euklidischen Abstände ein, sondern Diffusionsabstände. Weil die Verhältnisse dieser verschiedenen Abstände von nahe aneinander gelegenen Punkten aber ähnlich und Linearkombinationen

invariant unter Skalierungen sind, funktioniert der Ansatz der Rückabbildung auch bei Einbettungen mit Diffusion Maps.

3.2. Zur Lösung des Optimierungsproblems

Es handelt sich bei (3.3) um ein Optimierungsproblem mit linearer Restriktion, welches mangels Informationen über die Zielfunktion einen iterativen Ansatz erfordert [GK02]. Eine Umformulierung des Problems in eine geschlossene Form mit Hilfe der Lagrange-funktion erbringt ein handliches lineares Gleichungssystem [Fra15; Fra+14]. Beweise in diesem Sinne folgendes Lemma:

Lemma 3.1. *Das Optimierungsproblem der Rückabbildung (3.3) ist konvex.*

Beweis. Konstruiere die Grammatrix $G \in \mathbb{R}^{k \times k}$ mit den Einträgen $G_{ij} = (y^* - y^i)^T (y^* - y^j)$ für $i, j \in \mathcal{J}$ und forme die Kostenfunktion ohne Regularisierung auf (3.1) wie folgt um:

$$\begin{aligned} \left\| y^* - \sum_{j \in \mathcal{J}} \omega_j y^j \right\|_2^2 &= \left\| \sum_{j \in \mathcal{J}} \omega_j (y^* - y^j) \right\|_2^2 \\ &= \sum_{m=1}^p \left(\sum_{j \in \mathcal{J}} \omega_j (y_m^* - y_m^j) \right)^2 \\ &= \sum_{i, j \in \mathcal{J}} \omega_i \omega_j (y^* - y^i)^T (y^* - y^j) \\ &= \omega^T G \omega \end{aligned} \tag{3.5}$$

Unter Verwendung der Diagonalmatrix $C = \text{diag}(c_1, \dots, c_k)$ mit positiven Einträgen vereinfacht sich der Regularisierungsparameter $s : \mathbb{R}^k \rightarrow \mathbb{R}$ zu

$$\begin{aligned} s(\omega) &:= \|\omega\|_c^2 := \sum_{j \in \mathcal{J}} c_j \omega_j^2 \\ &= \omega^T C \omega. \end{aligned}$$

Integriere dies in die Umformulierung der Zielfunktion 3.5 mit $\tilde{G} = G + C$ und erhalte

$$\begin{aligned} \left\| y^* - \sum_{j \in \mathcal{J}} \omega_j y^j \right\|_2^2 + \|\omega\|_c^2 &= \omega^T G \omega + \omega^T C \omega \\ &= \omega^T \tilde{G} \omega \\ &=: f(\omega) \end{aligned} \tag{3.6}$$

mit $f : \mathbb{R}^k \rightarrow \mathbb{R}$ anstelle der Kostenfunktion in (3.3) [Fra15]. Da Grammatrizen und Diagonalmatrizen mit positiven Einträgen positiv semidefinit sind [Fis08], ist f konvex. Zudem liegt als Restriktion lediglich die lineare Gleichheitsnebenbedingung vor:

$$\sum_{j \in \mathcal{J}} \omega_j = \mathbb{1}^T \omega = 1.$$

Somit ist (3.3) äquivalent zu dem konvexen Optimierungsproblem [GK02]

$$\begin{aligned} \min \quad & f(\omega) \\ \text{u.d.N.} \quad & \mathbb{1}^T \omega = 1 \end{aligned} \tag{3.7}$$

mit $\omega \in \mathbb{R}^k$. □

Wir wissen, dass es eine eindeutige Lösung konvexer Optimierungsprobleme gibt und diese angenommen wird von einem ω^* , an dem der Gradient der Lagrangefunktion null ist [UU12].

Definiere die Lagrangefunktion $\mathcal{L} : \mathbb{R}^k \times \mathbb{R} \rightarrow \mathbb{R}$ zu dem restringierten konvexen Optimierungsproblem (3.7)

$$\mathcal{L}(\omega, \lambda) := \omega^T \tilde{G} \omega + \lambda(\omega^T \mathbb{1} - 1)$$

und berechne ihre Gradienten

$$\begin{aligned} \Delta_\omega \mathcal{L}(\omega, \lambda) &= 2\tilde{G}\omega + \lambda \mathbb{1} \\ \Delta_\lambda \mathcal{L}(\omega, \lambda) &= \mathbb{1}^T \omega - 1. \end{aligned}$$

Die notwendige und hinreichende Bedingung $\Delta \mathcal{L}(\omega, \lambda) = 0$ impliziert

$$\left\{ \begin{array}{l} \Delta_\omega \mathcal{L}(\omega, \lambda) = 2\tilde{G}\omega + \lambda \mathbb{1} = 0 \\ \Delta_\lambda \mathcal{L}(\omega, \lambda) = \mathbb{1}^T \omega - 1 = 0 \end{array} \right\} \Leftrightarrow \begin{pmatrix} 2\tilde{G} & \mathbb{1} \\ \mathbb{1}^T & 0 \end{pmatrix} \begin{pmatrix} \omega \\ \lambda \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \tag{3.8}$$

Dabei entspricht das Gleichsetzen von $\Delta_\lambda \mathcal{L}(\omega, \lambda)$ mit null der Einhaltung der Restriktion und von $\Delta_\omega \mathcal{L}(\omega, \lambda)$ der Annahme eines Minimums der Zielfunktion f aus (3.6).

Mit der geschlossenen Form (3.8) des Optimierungsproblems kann nun der Algorithmus „Rückabbildung“ wie folgt durchgeführt werden:

Algorithmus 3 : Rückabbildung

Data : $p^* \in \mathcal{P}$, $\Omega \subset \mathcal{P}$, $\mathcal{X}$, $\mathcal{X}_{red}$, k , ϵ , l

Result : Approximation x^* zu $f_{sim}(p^*) \in \mathbb{R}^d$

begin

$y^* = f_{trans}(p^*)$

$\mathcal{J}$ = Indizes der k nächsten Nachbarn von y^* in $\mathcal{X}_{red}$

 Initialisiere $G = [0]_{i,j=1}^k$, $C = [0]_{i,j=1}^k$

forall i, j **in** $\mathcal{J}$ **do**

$G_{ij} = (y^* - y^i)^T (y^* - y^i)$

$C_{ii} = c_i$

$\tilde{G} = G + C$

$[\omega, \lambda] \leftarrow \text{solveLGS} \left(\begin{bmatrix} 2\tilde{G} & \mathbb{1} \\ \mathbb{1}^T & 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right)$

 Initialisiere $x^* = [0]_{i=0}^k$

forall j **in** $\mathcal{J}$ **do**

$x^* = x^* + \omega_j x^j$

return x^*

3.3. Aufwand

r	Dimension von $\mathcal{P}$
p	Dimension von $\mathcal{X}_{red}$
k	Größe der Nachbarschaft
N	Anzahl der betrachteten Datenpunkten
d	Dimension von $\mathcal{X}$

Tabelle 3.3.: Natürliche Faktoren des Aufwandstermes

Tabelle 3.3 stellt übersichtlich alle Faktoren dar, die bei der Berechnung des Aufwandes der Rückabbildung benötigt werden. d ist dabei die mit Abstand größte Dimension. Außerdem kann die Samplegröße N je nach Anwendung einen mit p vergleichbaren Wert annehmen

Lemma 3.2. *Die Rückabbildung unter Verwendung einer RBF-Interpolation im ersten Teil und der LLI im zweiten Teil hat eine asymptotische Laufzeit von*

$$\mathcal{O}(pN(N^2 + rN) + k(\log N + k^2 + d)).$$

Beweis. Separate Betrachtung des Aufwandes beider Phasen der Rückabbildung (vgl. Tabelle 3.2) ergibt:

1. Angenommen, dass für die Interpolation in dem Einbettungsraum mit den Stützstellen in Ω eine RBF-Interpolation koordinatenweise verwendet wird, beträgt der Gesamtaufwand des ersten Abschnittes $\mathcal{O}(p(rN^2 + N^3 + N))$ [Fra15; Fra+14; SW06]; jede Interpolation in p Stellen erfordert:
 - a) das Erstellen der Kern-Matrix ($\mathcal{O}(rN^2)$),
 - b) das Lösen eines Linearen Gleichungssystems zum Bestimmen der Gewichte ($\mathcal{O}(N^3)$) und
 - c) die Anwendung der N Gewichte auf die dazugehörigen Samples unter einem Aufwand von $\mathcal{O}(N)$.
2. Für die Inversion der Dimensionsreduktion ergibt sich:
 - a) Um die k nächsten Nachbarn von y^* zu finden, müssen zuerst die euklidischen Abstände zu N Samples berechnet ($\mathcal{O}(pN)$) und anschließend die k kleinsten Abstände ausfindig gemacht werden, was eine Laufzeit von $\mathcal{O}(k \log N)$ hat [Hoa62].
 - b) Die Lösung des Gleichungssystems (3.8) setzt einen Aufwand von $\mathcal{O}((k+1)^3) = \mathcal{O}(k^3)$ voraus [Mei11].
 - c) Die Anwendung der Linearkombination aus $\mathcal{X}_{red}$ in $\mathcal{X}$ erfordert erstmalig den Umgang mit Daten aus dem hochdimensionalen Raum. Dieser besteht jedoch lediglich aus einer Skalarmultiplikation für jeden der k nächsten Nachbarn, sodass der Aufwand bei $\mathcal{O}(kd)$ liegt.

Summation der einzelnen Terme liefert eine Laufzeit von

$$\mathcal{O}(p(rN^2 + N^3 + N) + Np + k \log N + k^3 + kd).$$

□

Im Vergleich dazu hat eine direkte RBF-Interpolation in $\mathcal{X}$ einen Aufwand von

$$\mathcal{O}(d(rN^2 + N^3 + N)).$$

In der Praxis sind generell $N, d \gg r, p, k$. Unter dieser Voraussetzung lässt sich die Laufzeit der Rückabbildung mit dem wesentlichen Term

$$\mathcal{O}(pN^3)$$

und die der RBF-Interpolation mit

$$\mathcal{O}(dN^3)$$

zusammenfassen.

Im Optimalfall gelingt somit vermöge Verwendung der Rückabbildung eine Verringerung der Berechnungskomplexität um den Faktor $1 - \frac{p}{d}$.

Die später folgende Anwendung auf Simulationsdaten bestätigt diese Einschätzung; mit $p = 3$ und $d = 1000$ ist die Laufzeit der RBF-Interpolation im Beispiel POLY3 von 0.41737s um ca. 96% auf 0.01597s bei Benutzung der Rückabbildung reduziert. Verglichen mit der prognostizierten, theoretisch möglichen Beschleunigung um 99,7% werden damit die Vorteile der Rückabbildung deutlich, insbesondere im Hinblick darauf, dass mit einem unwesentlichen Stichprobenumfang von $N = 40$ Terme geringerer Ordnung zur Geltung kommen.

Eine direkte Anwendung des LLI von $\mathcal{P}$ nach $\mathcal{X}$ mit einer Laufzeit von

$$\mathcal{O}(Nr + k \log N + k^3 + kd)$$

ist dagegen effizienter als die Rückabbildung, welche die Daten aus dem Einbettungsraum verwendet.

4. Anwendungen

Zunächst werden zwei überschaubare, synthetische Anwendungen ohne Rauschen analysiert, deren intrinsische Struktur sehr zugänglich ist. Dies erlaubt, die Funktionsweise der Rückabbildung sowie ihre möglichen Schwächen am Beispiel greifbar aufzuarbeiten und zu diskutieren. Besonderes Augenmerk liegt dabei auch auf den Zusammenhängen zwischen Parameterraum und entstehender Einbettung. Sind diese Konzepte im zugänglichen Kontext erhellt, ist eine tiefere Analyse ihrer Auswirkungen im Rahmen zweier umfassender, realer Anwendungen angebracht. An dieser Stelle kann eine konkrete Einschätzung des praktischen Potentials der Rückabbildung vorgenommen werden. Einleitend nun eine Übersicht der untersuchten Modelle:

POLY2 Im Fokus steht das Beispiel (2.1) aus dem Abschnitt zu Isomap. Die Samples entstehen durch Evaluierung von 40 Polynomen zweiten Grades der Form $P_{p_1, p_2}(x) = p_1x + p_2x^2$ unter Variation der Koeffizienten p_1 und p_2 .

POLY3 Als Ergänzung zu POLY2 werden drei Koeffizienten von Polynomen dritten Grades der Gestalt $P_{p_1, p_2, p_3}(x) = p_1x + p_2x^2 + p_3x^3$ verändert (Abbildung 4.1). Durch die Erhöhung der Komplexität offenbaren sich neue Eigenarten der Rückabbildung.

PKW-CRASH Im Mittelpunkt steht simulierte Verformung des Querschnittes der PKW-Stirnwand eines Toyota Yaris, dessen Modell das NCAC¹ veröffentlicht hat. Konkret werden 234 Momentaufnahmen von verschiedenen Frontalcrashes betrachtet. Bei den Simulationen, welche mit LSDyna durchgeführt wurden, variiert die Winkelposition der Stoßfänger zwischen 0° und 360°. Die Abbildung 4.3 erläutert den Versuchsablauf [IG].

ZUGVERSUCH Es werden Spannungs-Dehnungs-Diagramme aus 2000 Zugversuchen analysiert, bei welchen eine Metallplatte erst auf 102% der Originallänge ausgedehnt und anschließend auf 98% gestaucht wird (siehe Abbildungen 4.2, 1.1a). Die Simulationen wurden mit LSDyna unter Verwendung des Yoshida-Uemori Material Model und Variation von 13 verschiedenen Parametern erstellt. Thyssenkrupp Steel² hat den Datensatz freundlicherweise für die Tests zur Verfügung gestellt.

Die Dimensionen der verschiedenen Räume und die Größe der Datensätze sind in Tabelle 4.1 aufgeführt.

¹Finite Element Model Archive, <http://www.ncac.gwu.edu/vml/models.htm> (2015)

²Thyssenkrupp Steel North America, <https://www.thyssenkrupp-north-america.com>

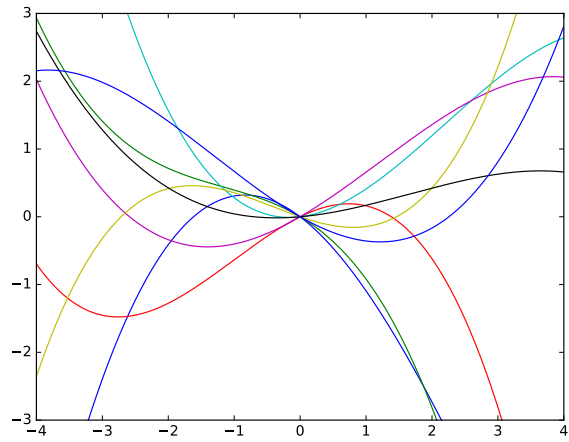


Abbildung 4.1.: Evaluierte Polynome des synthetischen Beispiels POLY3

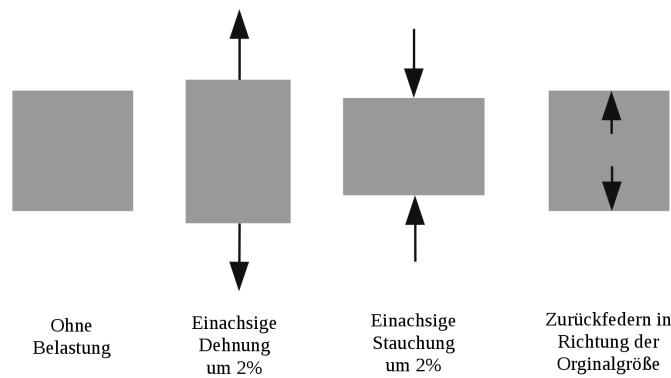
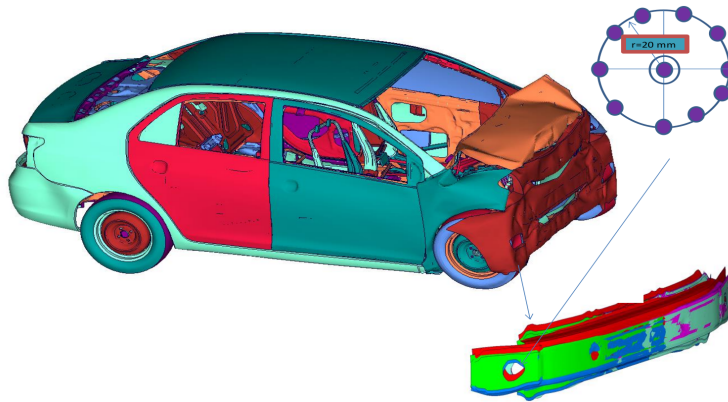


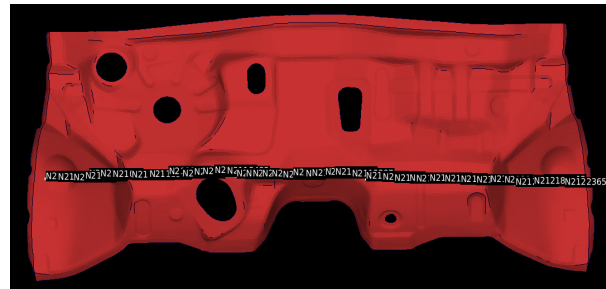
Abbildung 4.2.: Ablauf der Simulationen des Beispiels ZUGVERSUCH

Beispiel	Samplegröße N	r	Dimensionen	
			p	$d = t \cdot m$
POLY2	40	2	2	1000 = 1 · 1000
POLY3	40	3	3	1000 = 1 · 1000
PKW-CRASH	234	1	3	126 = 3 · 42
ZUGVERSUCH	2000	13	3	2000 = 2 · 1000

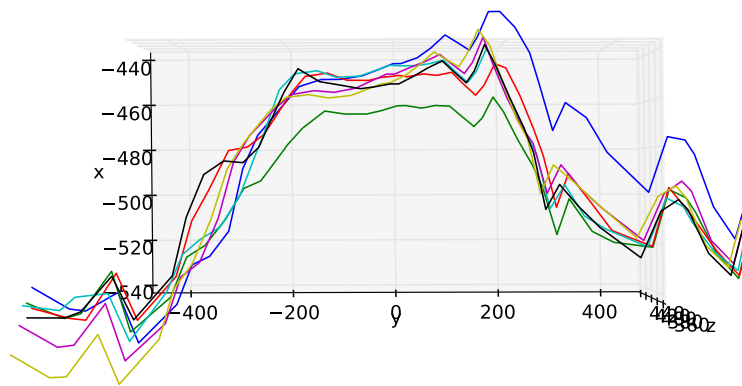
Tabelle 4.1.: Statistische Angaben betreffs verschiedener Datensätze



(a) Modell des Toyota Yaris nach dem Crash und Illustration der rotierenden Anbringung der Stoßfänger. Quelle: [IG]



(b) Beobachteter Querschnitt der PKW-Stirnwand



(c) Einige verformte Querschnitte der PKW-Wand am Ende des Crashes aus dem Raum $\mathcal{X}$

Abbildung 4.3.: Simulationen der Anwendung PKW-CRASH

4.1. Theoretische Aspekte der Analyse

Das Erstellen neuer Simulationen bei realen Anwendungen reagiert empfindlich auf Rauschen und ist zudem rechenaufwändig, weshalb zur Analyse der Rückabbildung schon vorhandene Samples einer Testmenge $\Omega_{Test} \subset \Omega$ approximiert werden. Für die Approximationen werden lediglich die übrigen Daten als bekannt vorausgesetzt. Diese Probe wird im Folgenden als Trainingsmenge $\Omega_{Train} = \Omega \setminus \Omega_{Test}$ bezeichnet. Zur Analyse der Inversion der Dimensionsreduktion $f_{red}^{[-1]}$ ist die Einbettung $\mathcal{X}_{red}$ in Test- und Trainingsmenge partitioniert.

Falls sich die Simulationen aus Ergebnissen an m verschiedenen Gitterpunkten zusammensetzen oder zu m verschiedenen Zeitpunkten erfasst wurden, besteht eine Simulation aus m t -dimensionalen Vektoren $(x^i)_1, \dots, (x^i)_m \in \mathbb{R}^t$ mit $i = 1, \dots, N, t \in \mathbb{N}$. Dann wird Isomap an den Vektoren

$$x^i = \begin{pmatrix} (x^i)_1 \\ \vdots \\ (x^i)_t \end{pmatrix} \in \mathbb{R}^{tm}$$

durchgeführt und es ist $d = t \cdot m$.

Isomap bettet so die Abstände

$$d_E^2(x^i, x^j) = \|x^i - x^j\|_2^2 = (x^i)^T(x^i) + (x^j)^T(x^j) - 2(x^i)^T(x^j)$$

ein [Boh+13]. Jedoch werden zum Vergleich von Datenpunkten die Abstände an den einzelnen Knoten

$$\begin{aligned} d_K(x^i, x^j) &= \|x^i - x^j\|_K \\ &= \frac{1}{m} \sum_{a=1}^m d_E((x^i)_a, (x^j)_a) \\ &= \frac{1}{m} \sum_{a=1}^m \sqrt{((x^i)_a - (x^j)_a)^T((x^i)_a - (x^j)_a)} \\ &= \frac{1}{m} \sum_{a=1}^m \sqrt{((x^i)_a)^T((x^i)_a) + ((x^j)_a)^T((x^j)_a) - 2((x^i)_a)^T((x^j)_a)} \end{aligned}$$

betrachtet. Es gilt aber, dass

$$d_E^2(x^i, x^j) = m \cdot d_{K,sq}(x^i, x^j) = \sum_{a=1}^m d_E^2((x^i)_a, (x^j)_a),$$

weshalb bei der Analyse dieser quadratische Fehler untersucht wird, da er zudem weniger aufwändig zu berechnen ist.

Beispiel	Durchschnittlicher Betrag $\mu = \frac{1}{N} \sum_i \ x^i\ _2^2$	Maximale Distanz $\eta = \max_{i,j} \ x^i - x^j\ _2^2$	$\frac{\eta}{\mu}$
POLY2	252	2137	8.48
POLY3	1936	4277	2.209
PKW-CRASH	23085300	96454	0.0042
ZUGVERSUCH	436846778	1108045828	2.5365

Tabelle 4.2.: Angaben zur Interpretation relativer Fehler bei Approximationen

Im vorstehenden Paragraphen wurden verschiedene absolute Fehler

$$\epsilon_{abs} = |x - x_{approx}|$$

eingeführt. Um diese in den Kontext einordnen zu können, werden bei den Analysen vorzugsweise relative Fehler

$$\epsilon_{rel} = \frac{|x - x_{approx}|}{|x|}$$

betrachtet, welche mittels Division des absoluten Fehlers durch die Norm von x entstehen. Der Wert ϵ_{rel} beschreibt dann die Abweichung der Approximation von dem Original im Verhältnis.

Bedauerlicherweise drückt ein relativer Fehler in keinem Sinne die Streuung der Daten aus. Bei geringen Entfernungen der Samplepunkte voneinander ist der Schwellwert für den relativen Fehler einer zufriedenstellenden Approximationen natürlich geringer, als wenn die Datenpunkte weit voneinander entfernt sind. Deshalb ist ein direkter Vergleich der relativen Fehler zu zwei verschiedenen Anwendungen nicht möglich, sondern nur der zweier unterschiedlicher Operationen auf einer Anwendung.

In Tabelle 4.2 sind die Verhältnisse der maximalen Distanzen, welche zwischen zwei Datenpunkten liegen, zu den durchschnittlichen Beträgen der Datenpunkte angegeben. Da dieses im Falle der Anwendung PKW-CRASH vergleichsweise klein ist, sind bei den Approximationen der Simulationsergebnissen kleine relative Fehler zu erwarten.

Zudem kann es zu Missverständnissen kommen, da bei der Betrachtung der relativen Fehler für jeden Datenpunkt eine individuelle Normierung gewählt wird. So fallen bei Datenpunkten, welche einen kleineren Betrag haben, Abweichungen stärker ins Gewicht, als bei betragsgrößeren Samples. Dieser Fall spielt bei den in dieser Arbeit untersuchten Beispielen jedoch keine große Rolle.

4.2. Untersuchung

Die Analyse der Rückabbildung wird in vier Schritten durchgeführt. Dabei werden die verschiedenen Anwendungen gleichzeitig betrachtet, um ausgewählte Aspekte besser gegenüberstellen zu können.

1. Mit der Dimensionsreduktion wird die Grundlage des Verfahrens untersucht. Zunächst hat die Absicherung einer adäquaten Einbettung Priorität. Wie bereits in

Kapitel 2.2.1 demonstriert, sind in diesem Kontext die Eigenwerte der Distanzmatrix Isomaps entscheidend, da mit ihrer Hilfe Rückschlüsse auf die intrinsische Dimension getroffen werden. Sekundär sind die Hyperparameter der Dimensionsreduktionen festzulegen; bei Isomap die Nachbarschaftsgröße k und Diffusion Maps ϵ .

2. Voraussetzung einer erfolgreichen Anwendung von $f_{red}^{[-1]}$ ist die Dualität der Nachbarschaftsbeziehungen in $\mathcal{X}_{red}$ und $\mathcal{X}$. Erstere sind gänzlich geprägt von der Effektivität der Transformation f_{trans} . Somit ist ebendiese von großer Tragweite, und in diesem Sinne ist die Wahl einer geeigneten Methode zur Transformation von $\mathcal{P}$ nach $\mathcal{X}_{red}$ das Ziel dieses Abschnittes.
3. Nun wird die mittels LLI durchgeführte Inversion der Dimensionsreduktion $f_{red}^{[-1]}$ analysiert. Hervorgehoben werden dabei Einflüsse der von LLI verwendeten Nachbarschaftsgröße, aber auch die Gewichte der Linearkombinationen in Relation zu den Distanzen assoziierter Datenpunkte.
4. Schlussendlich wird die vollständige Rückabbildung $f_{red}^{[-1]} \circ f_{trans}$ betrachtet; das Zusammenspiel beider Phasen des Verfahrens, die daraus resultierende Robustheit gegenüber Datenfehlern sowie die relative Performanz der Rückabbildung im Vergleich zu alternativen Verfahren werden evaluiert. In Bezug auf die Datenfehler wird eine Verbesserung des Verfahrens vorgeschlagen.

4.2.1. Dimensionsreduktion

Isomap kann, wie schon im Abschnitt 2.2.1 gesehen, die Variationen der Parameter im Beispiel POLY2 sehr gut zusammenfassen. Die Ausgabe von Diffusion Maps ist sehr ähnlich. Beide Einbettungen sind im Anhang in den Abbildungen A.2 und A.1 graphisch dargestellt.

Die Einbettungen zu POLY3 (siehe Abbildungen A.3, A.4) lassen dagegen keine eindeutige Beziehung zu den Koeffizienten des linearen Terms erkennen. Sein Einfluss auf die Funktionswerte ist, verglichen mit den Termen zweiten und dritten Grades, weniger stark. Dieses Verhalten erschwert die Transformation f_{trans} , da diese auf eine Strukturanalogie von P und $\mathcal{X}_{red}$ angewiesen ist. Die Analyse wird aber dennoch fortgesetzt, da auch in der Praxis oft der Fall eintritt, dass nicht jede Koordinate im Einbettungsraum unmittelbar mit einem Parameter aus P korreliert. Diese Problematik wird in Kapitel 4.2.5 erneut aufgegriffen.

Im Beispiel PKW-CRASH hat der Parameterraum die Dimension 1, die intrinsische Dimension des Simulationsraumes und damit auch die Dimension des Einbettungsraumes ist dagegen 3, da die Verformung beim Crashtest je nach Einstellung des Befestigungswinkels auf drei verschiedene Art und Weisen geschieht (siehe Abbildung A.5).

Die Dimensionsreduktion mit Isomap liefert eine unvorteilhafte Einbettung, weil im Mittelpunkt eine Strukturierung der Datenpunkte fehlt und ebenso die Eigenwerte der Graphendistanzmatrix nur langsam abfallen (siehe Abbildung A.5b). Deshalb wird in den folgenden Schritten hauptsächlich eine Einbettung mit Diffusion Maps verwendet,

welche die Datenpunkte wesentlich besser nach den unterschiedlichen Eingaben sortiert (siehe Abbildung A.6).

Die mit Isomap und Diffusion Maps reduzierten Räume des Beispiels ZUGVERSUCH sind im Gegensatz zum 13-dimensionalen Parameterraum $\mathcal{P}$ nur 3-dimensional. Die Rückabbildung wird daher zusätzlich hinsichtlich des Aufwandes im Vergleich zu einer direkten Interpolation vom 13-dimensionalen Parameterraum in den 2000-dimensionalen Simulationsraum interessant.

Beide Dimensionsreduktionen filtern den Einfluss des Parameters **cb** ausgezeichnet, den Einfluss der Parameter **y** und **ea** gut heraus. Die Einbettung mit Diffusion Maps ist jedoch um einiges kompakter und strukturierter in der Darstellung, was daran liegt, dass der Algorithmus globalere Zusammenhänge erkennt, während sich Isomap durch Verwendung von Nachbarschaften sehr auf die lokalen Eigenschaften der Daten beschränkt, jedoch daher auch die Abhängigkeit vom Parameter **ea** besser erfasst. Vergleichende Betrachtung der Abbildungen A.7 und A.8 bestätigt diese Erwartung.

4.2.2. Transformation f_{trans}

Für die Analyse wird die Transformation auf jeden Datenpunkt in Ω angewandt und das Ergebnis mit dem zugehörigen Datenpunkt aus $\mathcal{X}_{\text{red}}$ verglichen, sodass zu jeder Anwendung N Testmengen $\Omega_{\text{Test},i} = \{p^i\}$, $i = 1, \dots, N$ mit den zugehörigen Trainingsmengen $\Omega_{\text{Train},i} = \Omega \setminus \mathcal{X}_{\text{red},\text{Test},i}$ erstellt werden.

Die Verwendung einer RBF-Interpolation mit dem thin-plate-Kern ($r^2 \log r$) als Transformation ist empfehlenswert, sofern die Korrelation der Einbettung mit den Parametern global ausgeprägt ist, denn dieser Kern verschwindet insbesondere für große Radien nicht. Deshalb ist diese Transformation bei den Beispielen POLY2, PKW-CRASH und ZUGVERSUCH so effektiv.

Für die Polynome 3. Grades im Beispiel POLY3 empfiehlt es sich dagegen einen Algorithmus zu verwenden, der sich auf lokale Zusammenhänge konzentriert, da in der Einbettung mit einigen Parametern keine Zusammenhänge zu erkennen sind. Eine RBF-Interpolation ausgestattet mit dem Gauss-Kern $\exp\left(\frac{-r}{\epsilon}\right)^2$ [Ped+11] sowie einer kleinen Skalierung ist eine naheliegende Wahl, denn dieser nimmt für weit voneinander entfernte Punkte einen vernachlässigbaren Wert an (für Skalierungen siehe Abbildungsverzeichnis). Sie erzielt genauere Approximationen.

Eine weitere Möglichkeit für die Transformation von P nach $\mathcal{X}_{\text{red}}$ ist die Verwendung der LLI aus dem zweiten Teil der Rückabbildung, da diese ausschließlich auf lokale lineare Zusammenhänge anspricht. Bei den Beispielen POLY3, PKW-CRASH, sowie POLY2 in Kombination mit Isomap ist diese zudem recheneffiziente Approximation überlegen. Es wurde hier als Standardeinstellung die Nachbarschaftsgröße $k = 9$ mit $l = 4$ und $\epsilon = 0.01$ gewählt.

In der Abbildung 4.4 sind die Approximationsfehler der verschiedenen Methoden zusammengefasst und in Tabelle 4.3 die Ergebnisse der jeweils besten Methoden aufgeführt.

Bei einer Anwendung der Rückabbildung empfiehlt sich, die Transformation zunächst separat anzuwenden. Möglicherweise misslingt die Transformation und die Approximation

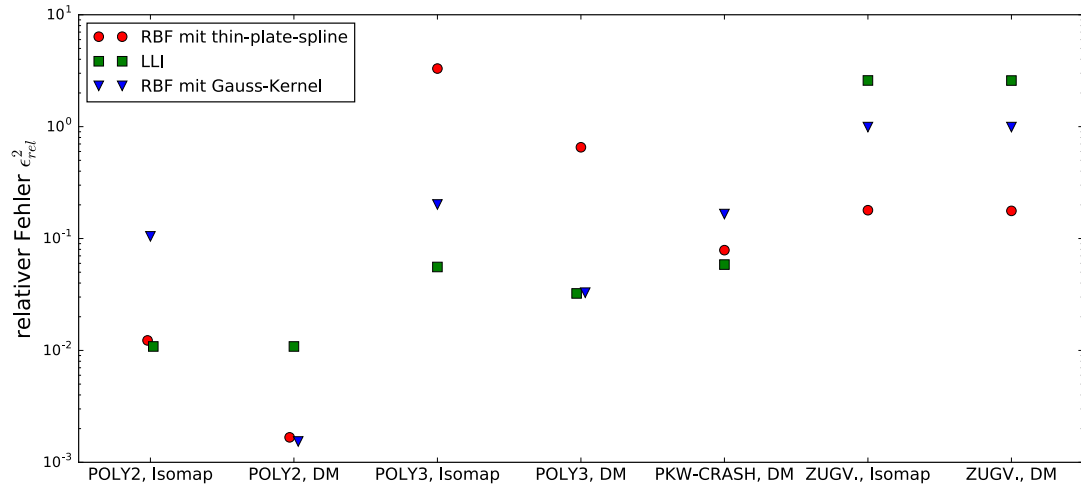


Abbildung 4.4.: Arithmetisches Mittel der relativen Fehler $\epsilon_{rel}^2 = \frac{1}{\|y^i\|_2^2} \cdot \|f_{trans}(p^i) - y^i\|_2^2$, $i = 1, \dots, N$ für verschiedene Wahlen der Transformation f_{trans}

Beispiel	Einbettung	Interpolation-Methode	$\emptyset$	ϵ_{rel}^2 σ	max
POLY2	Isomap	LLI	0.01085	0.01616	0.08567
	Isomap	Thinplate-RBF	0.01229	0.01453	0.05890
	Diff. Maps	Gauss-RBF	0.00154	0.00413	0.02496
	Diff. Maps	Thinplate-RBF	0.00167	0.00504	0.03155
POLY3	Isomap	LLI	0.05569	0.06528	0.26832
	Diff. Maps	LLI	0.03231	0.05376	0.27158
	Diff. Maps	Gauss-RBF	0.03281	0.08059	0.37445
PKW-CRASH	Diff. Maps	LLI	0.0585	0.09581	0.42165
ZUGVERSUCH	Isomap	Thinplate-RBF	0.17910	0.30111	2.10189
	Diff. Maps	Thinplate-RBF	0.17665	0.23279	1.36740

Tabelle 4.3.: Relative Fehler ϵ_{rel}^2 für jeweils besten Wahlen der Transformation f_{trans}

liegt außerhalb der eingebetteten Punkte. In diesem Fall sollte von einer vollständigen Anwendung der Rückabbildung abgesehen werden.

4.2.3. Inversion der Dimensionsreduktion $\mathbf{f}_{\text{red}}^{[-1]}$ mit LLI

An dieser Stelle stehen in den jeweiligen Beispielen die Güten der Dimensionsreduktionen fest. Daher ist es nun möglich, mit der Überprüfung der Inversion der Dimensionsreduktion fortzufahren. Zweckmäßig ist die Erstellung der Testmengen $\mathcal{X}_{\text{red},\text{Test},i} = \{y^i\}$, $i = 1, \dots, N$ mit den zugehörigen Trainingsmengen $\mathcal{X}_{\text{red},\text{Train},i} = \mathcal{X}_{\text{red}} \setminus \mathcal{X}_{\text{red},\text{Test},i}$, um den Einfluss der LLI-Nachbarschaftsgröße aufzudecken. Die Feinjustierung der Regularisierungsparameter ϵ und l wird im Rahmen der Betrachtung der vollständigen Rückabbildung vorgenommen.

Nachbarschaftsgröße

Im Allgemeinen fällt der relative Fehler $\epsilon_{\text{rel}}^2 = \frac{1}{\|x^i\|_2^2} \|f_{\text{red}}^{[-1]}(y^i) - x^i\|_2^2$ für $i = 1, \dots, N$ bei steigender Nachbarschaftsgröße, da bei unzureichender Anzahl von Nachbarn nicht genügend Informationen verwendet werden, um eine brauchbare Approximation des Simulationsergebnisses zu erlangen. Mit steigender Nachbarschaftsgröße fällt der Fehler langsamer (siehe Abbildung 4.5), da die Gewichtung der zunehmend weiter entfernten Nachbarn in der Linearkombination exponentiell kleiner wird. Dies wird besonders am Beispiel der Anwendung ZUGVERSUCH in Abbildung 4.6 deutlich.

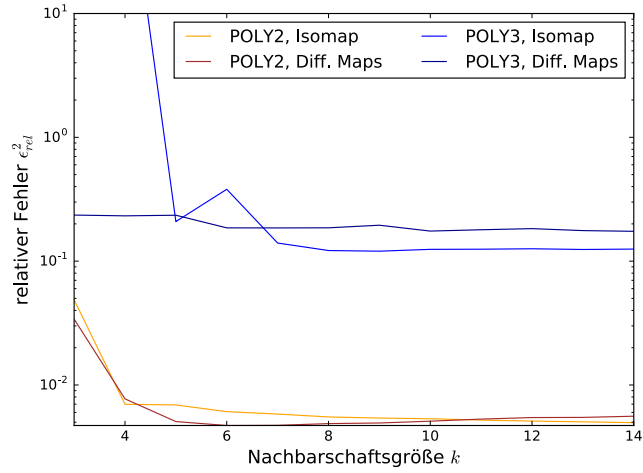
Die Anwendung der LLI auf die mit Diffusion Maps reduzierten Beispiele ist geglückt, obwohl der Algorithmus ad hoc keine euklidischen Abstände einbettet, auf welche die Inversion hingegen beruht. Wie in 3.1 begründet, sind die Verhältnisse der Diffusion Distances und der euklidischen Abstände untereinander bei kleinen Entfernungen ähnlich. Da die maximalen Fehler der LLI bei Diffusion Maps, welche in Tabelle 4.4 gelistet sind, um einiges geringer sind, werden diese vielversprechenden Einbettungen letztendlich auch bei der Analyse der Rückabbildung verwendet.

Bei der Anwendung PKW-CRASH steigt der Approximationsfehler ab Nachbarschaftsgröße 7 wieder. Das ist eine Konsequenz aus der unzureichenden Einbettung mit Isomap, welches Schwierigkeiten mit der Einbeziehung weit entfernter Nachbarn zeigt. Deshalb wird die Anwendung PKW-CRASH nicht mehr in Kombination mit Isomap betrachtet.

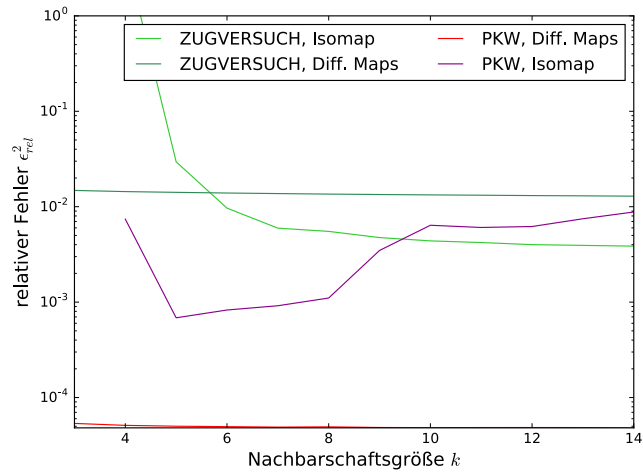
Zusammenfassend lässt sich also in Bezug auf die hier getesteten Samples sagen, dass unter der Voraussetzung einer erfolgreichen Dimensionsreduktion ihre Inversion mit LLI sehr zufriedenstellende Approximationsergebnisse liefert. Alle Angaben zu den relativen Fehlern mit der jeweils besten Nachbarschaftsgröße sind in Tabelle 4.4 vermerkt.

4.2.4. Vollständige Rückabbildung $\mathbf{f}_{\text{red}}^{[-1]} \circ \mathbf{f}_{\text{trans}}$

Mit der vollständigen Rückabbildung werden die Simulationsergebnisse zu Parameter-einstellungen aus zufällig gewählten Testmengen Ω_{Test} approximiert. Hierzu werden nur die Daten der Trainingsmengen verwendet, das heißt insbesondere, dass bei der



(a) Synthetische Beispiele



(b) Reale Beispiele

Abbildung 4.5.: Durchschnittliche relative Fehler $\epsilon_{rel}^2 = \frac{1}{\|x^i\|_2^2} \|f_{red}^{[-1]}(y^i) - x^i\|_2^2, i = 1, \dots, N$ der unterschiedlichen Beispiele für verschiedene Nachbarschaftsgrößen $k = 3, \dots, 14$ bei der Inversion der Dimensionsreduktion mit LLI

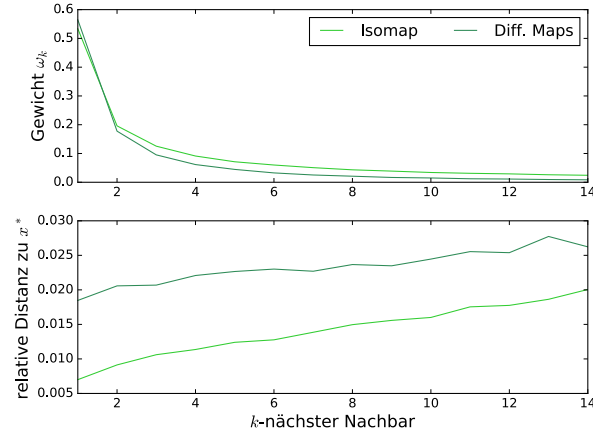


Abbildung 4.6.: Durchschnittliche Gewichte der 14 nächsten Nachbarn bei den Linearkombination und Entfernungen dieser Nachbarn in $\mathcal{X}$ für die Anwendung Zugversuch

Beispiel	Einbettung	k	relative Fehler ϵ_{rel}^2		
			$\emptyset$	σ	max
POLY2	Isomap	9	0.00541	0.00780	0.97889
	Diffusion Maps	7	0.00472	0.00797	0.44938
POLY3	Isomap	8	0.12184	0.23927	6298.32532
	Diffusion Maps	10	0.17495	0.29294	5.11232
PKW-CRASH	Isomap	5	0.00069	0.00586	1.61241
	Diffusion Maps	7	0.00005	0.00006	0.00067
ZUGVERSUCH	Isomap	9	0.00475	0.01438	2173.60993
	Diffusion Maps	9	0.01342	0.02666	0.34979

Tabelle 4.4.: Arithmetische Mittel, Standardabweichungen und maximale Werte der relativen quadratischen Fehler ϵ_{rel}^2 für vorläufig gewählte Nachbarschaftsgrößen bei der Inversion der Dimensionsreduktion mit LLI

Beispiel	Anzahl der Testmengen	Größe der Testmengen	Größe der Trainingsmengen
POLY2	1	20	40
POLY3	1	20	40
PKW-CRASH	3	10	224
ZUGVERSUCH	2	20	1980

Tabelle 4.5.: Wahl der Mengen Ω_{Test} und Ω_{Train} zur Analyse der vollständigen Rückabbildung

Dimensionsreduktion die den $p \in \Omega_{Test}$ zugehörigen tatsächlichen Simulationsergebnisse ausgeschlossen wurden. So ist es möglich, diese abschließend zur Evaluierung der Performanz mit der Berechnung der relativen Fehler heranzuziehen.

Anzahl und Umfang der jeweiligen Test- und Trainingsmengen sind in Tabelle 4.5 vermerkt.

Zu Beginn der Analyse der kompletten Rückabbildung wird die optimale LLI-Nachbarschaftsgröße k für jedes Beispiel in Abhängigkeit der verschiedenen Transformationen bestimmt (siehe Abbildung 4.7), um danach einen konkurrenzfähigen Vergleich zu einer direkten Interpolation in $\mathcal{X}$ durchzuführen. Beispielsweise liefert eine direkte RBF-Interpolation ungenügende Ergebnisse, da sie komponentenweise durchgeführt werden muss und ihr Einflüsse der verschiedenen Koordinaten aufeinander daher entgehen. Verhältnismäßig besser ist eine direkte LLI mit der Parametrisierung Nachbarschaftsgröße $k=9$, $l=4$ und $\epsilon = 0.01$, deren Ergebnisse in Tabelle 4.6 aufgelistet sind.

Im Falle der Anwendungen POLY2, POLY3 und PKW-CRASH liegen die Approximationsfehler der Rückabbildung in derselben Größenordnung wie die der nicht angepassten direkten LLI.

Bei den Beispielen zu den Polynomauswertungen ist das akzeptabel, da der Zusammenhang zwischen Koeffizienten und Graph des Polynoms vergleichsweise linear ist. Der Umweg über den Einbettungsraum ist bei dem Beispiel POLY2 eine unnötige Fehlerquelle und bei POLY3 nicht vorteilhaft, da die Einbettungen zu unstrukturiert sind. Im folgenden Abschnitt wird letztere Problematik durch gezielte Modifikation der Nachbarschaftskonstruktion weitgehend behoben.

Die Approximationsfehler zur Anwendung ZUGVERSUCH bei der Rückabbildung übertreffen die der direkten LLI um den Faktor 10. Der Defekt der direkten RBF-Interpolation wird vernachlässigt, da auf Grund einer hohen Laufzeit nur ein Sample getestet wurde.

4.2.5. Variation der Nachbarschaftswahl

Standardmäßig werden für die Auswahl der Linearkombinationen beim Invertieren der Dimensionsreduktion die nächsten Nachbarn von y^* aus dem Einbettungsraum X_{red} gewählt. Dies ist problematisch, wenn die Transformation $y^* = f_{trans}(p^*)$ nicht zufriedenstellend ist, da bei der Einbettung wichtige Informationen über den Zusammenhang

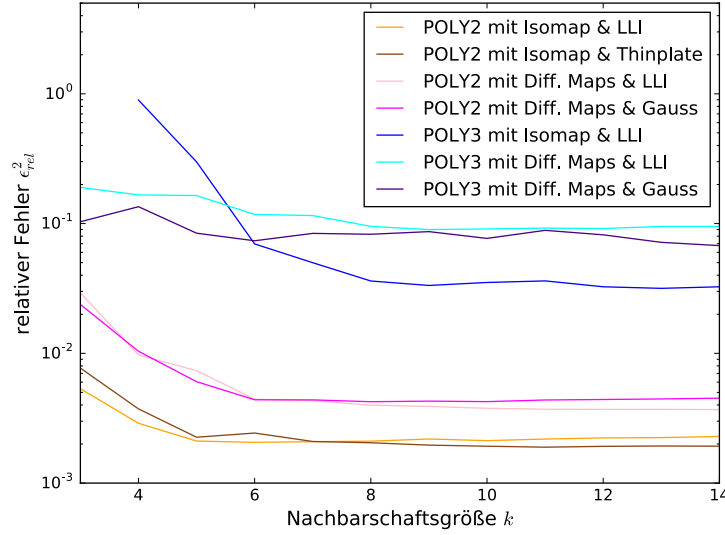


Abbildung 4.7.: Arithmetische Mittel der relativen Approximationsfehler $\epsilon_{rel}^2 = \frac{1}{\|x^i\|_2^2} \|f_{red}^{[-1]} \circ f_{trans}(p^i) - x^i\|_2^2$, $p^i \in \Omega_{Test}$ für verschiedene Nachbarschaftsgrößen k bei der kompletten Rückabbildung

Beispiel	direkte Interpolation		Rückabbildung			
	RBF	LLI	Einbettung	f_{trans}	k	Fehler
POLY2	0.00425	0.003416	Isomap	LLI	9	0.00219
POLY3	3.64519	0.019897	Isomap	LLI	8	0.036089
PKW-CRASH	0.000117	$8.776 \cdot 10^{-5}$	Diff. Maps	LLI	10	$6.902 \cdot 10^{-5}$
ZUGVERSUCH	0.15628*	0.51221	Isomap	RBF	10	0.044392

Tabelle 4.6.: Vergleich der Approximationsfehler ϵ_{rel}^2 der direkten Interpolationen in $\mathcal{X}$ und der Rückabbildung, welches zusätzliche Informationen aus $\mathcal{X}_{red}$ verwendet

* Aufgrund sehr hoher Laufzeit konnte die Approximation an lediglich einem Sample durchgeführt werden.

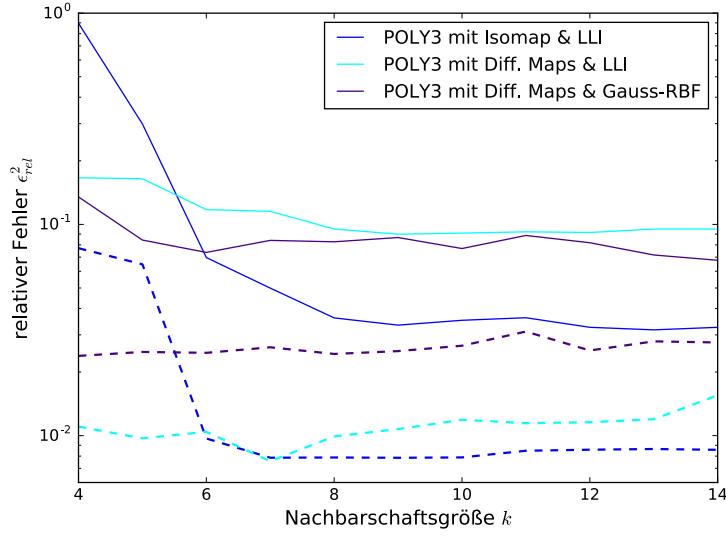


Abbildung 4.8.: Arithmetisches Mittel der relativen Fehler ϵ_{rel}^2 der Rückabbildung beim Beispiel POLY3. Nachbarn aus $\mathcal{X}_{red}$: —, Nachbarn aus $\mathcal{P}$: - - -

zwischen Parameterraum $\mathcal{P}$ und Simulationsraum $\mathcal{X}$ nicht zum Ausdruck kommen.

Folglich enthält die Nachbarschaft von y^* im Einbettungsraum $\mathcal{X}_{red}$ nicht genügend Informationen über das gesuchte Simulationsergebnis. Daher sollte sich eine Bewältigungsstrategie zu dieser Schwierigkeit den Mangel an Informationen über die Struktur von $\mathcal{P}$ vornehmen; die LLI im zweiten Teil der Rückabbildung wird stattdessen unter Verwendung der nächsten Nachbarn von p^* aus dem Parameterraum durchgeführt:

1.	Projiziere p^* in den Einbettungsraum.	$y^* = f_{trans}(p^*)$
2.a	Finde Indices $\mathcal{J}_{\mathcal{P}}$ der k nächsten Nachbarn von p^* in $\mathcal{P}$.	$\{y^j \mid j \in \mathcal{J}_{\mathcal{P}}\} \subset \mathcal{X}_{red}$
2.b	Berechne Gewichtsvektor $\omega^* \in \mathbb{R}^k$ zur Approximation von y^* im Einbettungsraum.	$\min_{\omega \in \mathbb{R}^k} \ y^* - \sum_{j \in \mathcal{J}_{\mathcal{P}}} \omega_j y^j\ _2^2 + \ \omega\ _c^2$ u.d.N. $\sum_{j \in \mathcal{J}} \omega_j = 1$
2.c	Benutze Gewichtsvektor zur Interpolation im Simulationsraum.	$x^* = \sum_{j \in \mathcal{J}_{\mathcal{P}}} \omega_j^* x^j$

Tabelle 4.8.: Variierte Rückabbildung in vier Schritten (Verwendung der nächsten Nachbarn aus $\mathcal{P}$)

In Bezug auf Anwendung POLY3 wurde bereits in Kapitel 4.2.1. erwähnt, dass die Dimensionsreduktionen nicht alle Abhängigkeiten des Parameterraumes widerspiegeln. Deshalb sind in diesem Fall die Ergebnisse der Transformation f_{trans} besonders unzurei-

Einbettung	f_{trans}	Nachbarschaft	k	relativer Fehler ϵ_{rel}^2		
				$\emptyset$	σ	max
Isomap	LLI	$\mathcal{P}$	7	0.00786	0.01950	0.08802
		$\mathcal{X}_{red}$	8	0.03609	0.05899	0.24035
Diff. Maps	LLI	$\mathcal{P}$	7	0.00757	0.01391	0.04858
		$\mathcal{X}_{red}$	9	0.08985	0.19044	0.82438
Diff. Maps	Gauss	$\mathcal{P}$	8	0.02436	0.03203	0.13566
		$\mathcal{X}_{red}$	6	0.07369	0.12319	0.51543

Tabelle 4.9.: Verbesserung der relativen Approximationsfehler ϵ_{rel}^2 durch die veränderte Nachbarschaftswahl beim Beispiel POLY3

chend, weshalb an diesem synthetischen Beispiel die Reaktion der Approximationsfehler auf Adaption der Nachbarschaftskonstruktion untersucht wird. Sowohl durchschnittlicher relativer Approximationsfehler, als auch dessen Standardabweichung und maximaler Wert verbessern sich maßgeblich, wenn Nachbarn aus dem Parameterraum verwendet werden. Dieses Ergebnis ist in Tabelle 4.9 und Graph 4.8 dargestellt.

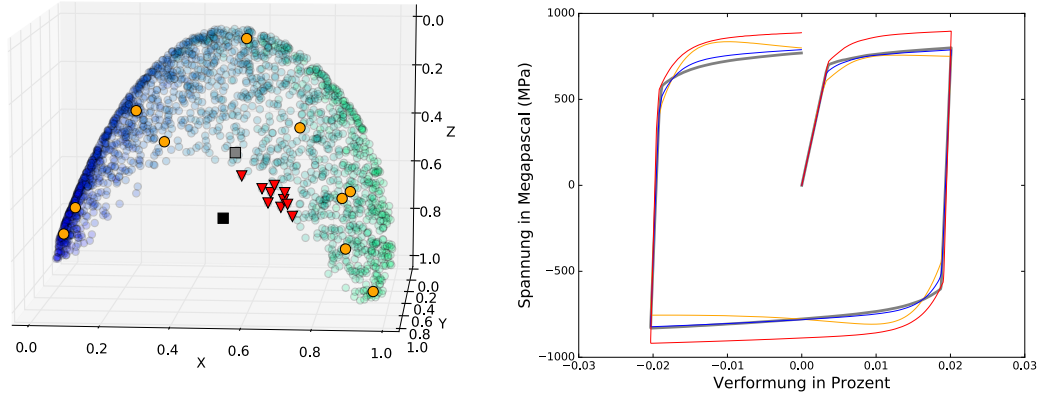
Des Weiteren konnte vermöge derselben Anpassung bei der Anwendung ZUGVERSUCH mit Diffusion Maps der Approximationsfehler verbessert werden. Dies scheint zunächst überraschend, da in diesem Kontext die Inversion der Dimensionsreduktion hervorragend gelingt und keine lokale, lineare Abhängigkeit zwischen $\mathcal{P}$ und $\mathcal{X}$ vorliegt. Die in hohem Maße fehlerbehaftete Approximation mittels direkter LLI in $\mathcal{X}$ ist ein starker Hinweis auf Letzteres.

Bei genauerer Betrachtung wird deutlich, dass die Herausforderung dieser Anwendung die bei der Transformation f_{trans} auftretenden großen Abweichungen ist, wie in Kapitel 4.2.2 herausgearbeitet. Daher stehen für den zweiten Teil der Rückabbildung lediglich verfälschte Eingaben zur Verfügung (vgl. Abbildung 4.9a). Die Verwendung der nächsten Nachbarn aus $\mathcal{P}$ wirkt dem entgegen. In dieser Hinsicht kann auch die Übereinstimmung euklidischer Abstände weit entfernter Punkte in $\mathcal{X}_{red}$ und $\mathcal{X}$, welche eine Besonderheit dieser Anwendung ist, zur Verbesserung der Resultate zu Nutze gemacht werden.

Regularisierungsparameter l und ϵ

Der Parameter l bestimmt, wie stark die Verwendung weiter entfernter Nachbarn für die Approximation bestraft wird. Dass es bei den betrachteten Beispielen relativ klein gewählt wird, hat seinen Grund in der Verwendung der Nachbarn aus dem Parameterraum, deren Abstände im Einbettungsraum eine hohe Standardabweichung haben. Große Abstände bedeuten aber nicht unbedingt schlechte Nachbarn, weshalb dieser Aspekt durch ein $l > 1$ potentiell unverhältnismäßig bestraft würde.

Weil bei den realen Anwendungen im Gegensatz zu den synthetischen Polynombeispielen verrauschte Datensätze vorliegen, sollte der Regularisierungsterm hier mehr ins Gewicht fallen. Deshalb ist die Approximation bei PKW-CRASH und ZUGVERSUCH mit großem ϵ , welches den Einfluss der Regularisierung bestimmt, exakter, während man



- (a) Einbettung mit Diffusion Maps. $\blacksquare$: Originaleinbettung, $\blacksquare$: mit RBF-Interpolation approximierte Einbettung, $\blacktriangledown$: nächste Nachbarn von x^* in $\mathcal{X}_{red}$, $\bullet$: nächste Nachbarn von p^* in P
- (b) Approximationen des originalen Spannungs-Dehnungs-Diagramm (grau). Orange: Verwendung der Nachbarn von p^* in P , rot: Verwendung der Nachbarn von x^* in $\mathcal{X}_{red}$, Blau: Ergebnis der Inversion der Dimensionsreduktion mit LLI unter Verwendung von $\hat{y}$

Abbildung 4.9.: Vergleich der Approximationen in $\mathcal{X}_{red}$ und $\mathcal{X}$ eines durchschnittlichen Beispiels der Anwendung ZUGVERSUCH

bei POLY2 und POLY3 für kleinere ϵ genauere Ergebnisse erzielt.

Die besten Einstellungen in den verschiedenen Phasen der Rückabbildung sind für alle Anwendungen in der Tabelle 4.10 aufgeführt. Die dazugehörigen relativen Approximationsfehler sind in Tabelle 4.11 zu finden, und in Abbildung 4.10 visualisiert.

4.2.6. Laufzeitvergleich

Ein Laufzeitvergleich der verschiedenen Interpolationsmethoden bestätigt, was in Kapitel 3.3 behauptet wurde. Wie in Tabelle 4.12 vermerkt, terminiert die Rückabbildung mehr als zehnmal schneller als eine direkte RBF-Interpolation, weil diese aufgrund der hohen Dimensionalität von $\mathcal{X}$ sehr aufwändig ist.

Die Laufzeiten der direkten Anwendung der LLI von $\mathcal{P}$ nach $\mathcal{X}$ und der Rückabbildung sind ähnlich und unterscheiden sich hauptsächlich in der verwendeten Nachbarschaftsgröße. Auch die Einbeziehung des bereits untersuchten relativen Fehlers in die Bewertung der Laufzeit schafft keine Klarheit; in beider Hinsicht sind direkte LLI und Rückabbildung durchweg vergleichbar und kein Verfahren ist dem anderen eindeutig überlegen.

Eine Ausnahme bilden die Zugversuche, in deren Kontext die Rückabbildung mit erheblich genaueren Resultaten bei komparabler Laufzeit punktet. Daher kann für diese Anwendung eine Empfehlung zu Gunsten der Rückabbildung ausgesprochen werden.

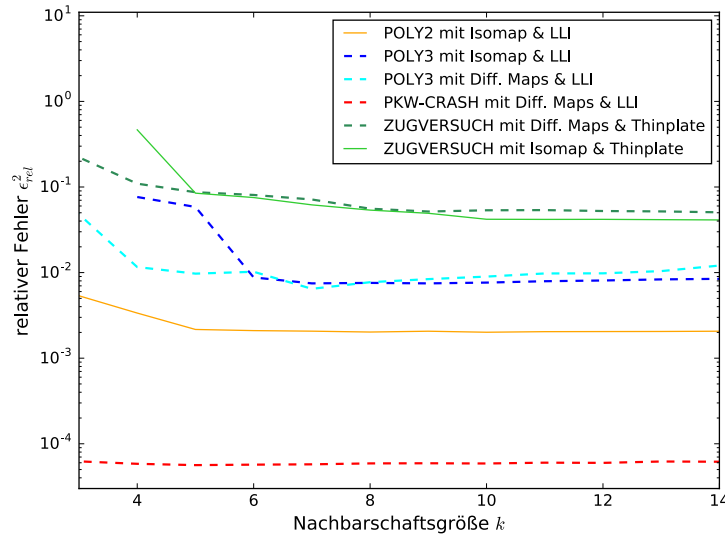


Abbildung 4.10.: Relativer Approximationsfehler $\epsilon_{rel}^2 = \frac{1}{\|x^i\|_2^2} \|f_{red}^{[-1]} \circ f_{trans}(p^i) - x^i\|_2^2$, $p^i \in \Omega_{Test}$ der Rückabbildung in Abhängigkeit der LLI-Nachbarschaftsgröße unter Verwendung der optimalen Einstellungen aus Tabelle 4.10

Anwendung	Einbettung	Einstellungen der Rückabbildung				
		1. Teil	Nachbar-schaft	k	l	ε
POLY2	Isomap	LLI	$\mathcal{X}_{red}$	9	5	0.01
POLY3	Diff. Maps	LLI	$\mathcal{P}$	7	2	0.01
	Isomap	LLI	$\mathcal{P}$	7	3	0.01
PKW-CRASH	Diff. Maps	LLI	$\mathcal{P}$	6	1	0.04
ZUGVERSUCH	Diff. Maps	Thinplate	$\mathcal{P}$	10	1	0.02
	Isomap	Thinplate	$\mathcal{X}_{red}$	10	1	0.04

Tabelle 4.10.: Die optimalen Einstellungen der Rückabbildung für alle Anwendungen

Anwendung	Einbettung	relativer Fehler ϵ_{rel}^2		
		∅	σ	max
POLY2	Isomap	0.0020657	0.0031735	0.010844
POLY3	Diff. Maps	0.0064617	0.011439	0.050483
	Isomap	0.0074804	0.018118	0.081789
PKW-CRASH	Diff. Maps	$5.703 \cdot 10^{-5}$	$5.9452 \cdot 10^{-5}$	0.00021723
ZUGVERSUCH	Diff. Maps	0.053449	0.091016	0.36525
	Isomap	0.041979	0.074642	0.40027

Tabelle 4.11.: Relative Approximationsfehler ϵ_{rel}^2 der Rückabbildung unter Verwendung der optimalen Einstellungen aus Tabelle 4.10

Beispiel	Methode		Laufzeit
POLY2	direkte Interpolation	LLI	0.01085
		RBF	0.56731
	Rückabbildung	LLI	0.00946
		RBF	0.00622
POLY3	direkte Interpolation	LLI	0.01097
		RBF	0.41737
	Rückabbildung	LLI	0.01530
PKW-CRASH	direkte Interpolation	LLI	0.00376
		RBF	0.41225
	Rückabbildung	LLI	0.00510
ZUGVERSUCH	direkte Interpolation	LLI	0.02995
		RBF	1598.65178*
Diffusion Maps	Rückabbildung	RBF	0.04355
Isomap	Rückabbildung	RBF	0.04367

Tabelle 4.12.: Laufzeiten der verschiedenen Methoden zur Approximation der Simulationsergebnisse.

*Aufgrund sehr hoher Laufzeit konnte die Approximation an lediglich einem Sample durchgeführt werden.

4.3. Ergebnisse

Zusammenfassend werden die verschiedenen Anwendungen mit den Ergebnissen der einzelnen Tests aufgeführt:

POLY2 & POLY3 Bei den Polynomen zweiten und dritten Grades kann die Rückabbildung eine direkte Interpolation weder in Effizienz noch Effektivität maßgeblich übertreffen.

Jedoch zeigt das Beispiel POLY3, wie die Wahl der nächsten Nachbarn aus $\mathcal{P}$ den schlechten Einfluss einer ungenügenden Dimensionsreduktion mindern und damit die Performanz des Verfahrens verbessern kann.

PKW-CRASH Bei dieser Anwendung wurde die Dimensionsreduktion mit Isomap durch eine exaktere mit Diffusion Maps ersetzt. Durch die Verwendung der nächsten Nachbarn aus dem Parameterraum beim Invertieren der Dimensionsreduktion kann sich die Rückabbildung gegenüber einer direkten Interpolation auszeichnen.

ZUGVERSUCH Es wurde hier eine niedrigdimensionale Einbettung sowohl mit Isomap als auch Diffusion Maps betrachtet. Beide erzielen mit der Rückabbildung weitaus exaktere Approximationen der Simulationsergebnisse als eine direkte Interpolation in $\mathcal{X}$. Die RBF-Interpolation hat zudem eine immense Laufzeit.

Da die Transformation f_{trans} von $\mathcal{P}$ in den Einbettungsraum von Diffusion Maps

Einbettung	relative Fehler ϵ_{rel}^2		
	direkte Interpolation	Inversion der Dimensionsreduktion	Rückabbildung
Diff. Maps	0.51221	0.01343	0.053449
Isomap	0.51221	0.00475	0.041979

Tabelle 4.13.: Relative Fehler der verschiedenen Approximationen am Beispiel ZUGVERSUCH

ungenau ist, empfiehlt sich die Verwendung der Nachbarschaften aus dem Parameterraum für die Inversion der Dimensionsreduktion.

Außerdem kann die Instabilität der ersten Phase der Rückabbildung wahrgenommen werden, welche ein Ausnutzen der ausgezeichneten Approximationsergebnisse von $f_{red}^{[-1]}$ verhindert. Die nichtsdestotrotz guten Ergebnisse der Rückabbildung sowie der Inversion der Dimensionsreduktion sind im Vergleich zu einer direkten Interpolation in $\mathcal{X}$ in Tabelle 4.13 vermerkt.

5. Resümee und Ausblick

Zusammenfassung

Mein Interesse an Verfahren zur Dimensionsreduktion gab die Richtung der Themenwahl meiner Bachelorarbeit an. Daher weckte der Vorschlag, mich der Inversion von Verfahren zur nichtlinearen Dimensionsreduktion zu widmen, sofort meine Neugier. Entgegen meiner ursprünglichen Erwartung, hauptsächlich eine solche inverse Abbildung zu untersuchen, ermöglichte die Auseinandersetzung mit den Inhalten durch den vorgegebenen Kontext der Autocrashtestsimulationen die Einarbeitung in mir neue, faszinierende Methoden.

So gelang die Implementierung der LLI, welche letztendlich zur Umkehrung der Dimensionsreduktionen Isomap und Diffusion Maps verwendet wurde, nach eingehender Beschäftigung mit den theoretischen Grundlagen. Insgesamt bin ich mit der Effektivität der lokalen linearen Interpolation sehr zufrieden; in allen vorliegenden Anwendungen wurde die in den Einbettungen repräsentierte intrinsische Struktur in hohem Maße genutzt.

Im Gegensatz dazu gestaltete sich die Kombination der Inversion $f_{red}^{[-1]}$ mit der Transformation f_{trans} im Rahmen der Rückabbildung als große Herausforderung. Die Tatsache, dass eine Korrelation der Parametereinstellungen und der eingebetteten Daten nicht zwangsläufig vorliegt, führt oftmals zu fehlerhaften Ausgaben $y^* = f_{trans}(p^*)$. An dieser Stelle sind somit Modifikationen der Rückabbildung erforderlich, die verschwundene Informationen aus dem Parameterraum aufspüren, damit die Umkehrung der Dimensionsreduktion bessere Eingaben erhält.

Mit dem Ansatz, sich von den nächsten Nachbarn einer mangelhaften Approximation y^* zu lösen und Nachbarn der Parametereinstellung p^* in $\mathcal{P}$ zu verwenden, konnte ich obiger Problematik vorbeugen.

In dieser Hinsicht wird eine tolle Eigenschaft der Rückabbildung deutlich: Durch die sequentielle Vorgehensweise ist sie vollständig modular. Auf Besonderheiten der Datensätze kann flexibel reagiert werden, indem ihre einzelnen Komponenten zielgerichtet adaptiert oder durch geeignetere Ansätze gänzlich ersetzt werden. Außerdem ist befriedigend, dass zur Analyse von Datensätzen oft ohnehin schon generierte Einbettungen der Rückabbildung dienen.

Auf diese Weise konnten bei einigen Anwendungen beeindruckende Resultate erzielt werden, die sich vor einem Vergleich mit gängigen Methoden zur direkten Interpolation in $\mathcal{X}$ nicht scheuen brauchen und eher naive Techniken, wie eine RBF-Interpolation koordinatenweise, in den Schatten stellen.

Bedenkt man, dass ich vor dieser Erörterung auf solche eindimensionale Handhabungen angewiesen war, ist einsichtig, dass mich diese Chance zur Entwicklung meines Repertoires sehr erfreut.

Ausblick

Aufgrund vielversprechender Ergebnisse beim ZUGVERSUCH sollte die Rückabbildung an weiteren Anwendungen untersucht werden. Besonders interessant dafür sind Datensätze, welche stabile Transformationen von dem Parameterraum in den Einbettungsraum ermöglichen. Um diese zu finden, müssen die Voraussetzungen des Erfolgs von f_{trans} noch genau erarbeitet werden.

Ferner macht die Bedeutsamkeit der Transformation eine effiziente Methode zur Bewertung ihrer Funktionalität erforderlich. Dabei sollte auch die Vergleichbarkeit der Resultate zwischen verschiedenen Applikationen Priorität haben.

An solchen Daten kann die Modifizierbarkeit der Rückabbildung ausgereizt werden; ein gutes Beispiel ist die adaptive Wahl der LLI-Nachbarschaftsgröße bei der Inversion der Dimensionsreduktion [Fra15]. Schlüsselgedanke dabei ist die Verwendung möglichst konvexer Linearkombinationen aus wenigen Punkten, um für jedes Sample individuell die optimale Anzahl der einbezogenen Nachbarn abzuschätzen.

Darüber hinaus ist die Untersuchung der Rückabbildung in Kombination mit Karcher means zur nichtlinearen Interpolation vom Parameterraum in den Einbettungsraum, wie in [TSK09] durchgeführt, beachtenswert. Auf diese Art und Weise kann möglicherweise die Transformation f_{trans} an Robustheit gewinnen.

Diskutiert diese Arbeit unter anderem die Vorhersage von Simulationsergebnissen, so hat eine Inversion der Dimensionsreduktion auch Anwendung ohne vorherige Transformation von $\mathcal{P}$ nach $\mathcal{X}_{red}$. Denkbar ist diesbezüglich der Einsatz zur Aufbereitung von Daten, die zunächst eingebettet und anschließend ohne Rauschen mittels LLI rekonstruiert werden. Dies wäre in der Bildverarbeitung von großem Wert.

All dies eingerechnet, halte ich das Potenzial sowohl der Inversion von Dimensionsreduktionen als auch im erweiterten Sinne der Rückabbildung für beträchtlich. Ich bin gespannt auf Weiterentwicklungen, welche die Zukunft bringt und hoffe, an diesen mitwirken zu können.

A. Plots der Dimensionsreduktionen

Es folgen die Darstellungen der Einbettungen mit Isomap und Diffusion Maps der verschiedenen Anwendungen. Im dreidimensionalen Fall sind zuerst ein dreidimensionaler Plot und anschließend Projektionen auf die Koordinatenebenen abgebildet, die gegebenenfalls nach einem Parameter der Anwendung eingefärbt sind. Die Darstellungen der Dimensionsreduktionen mit Isomap enthalten zusätzlich einen Plot der 10 größten Eigenwerte der Graphendistanzmatrix, aus welchen die intrinsische Dimension p abgeleitet wurde.

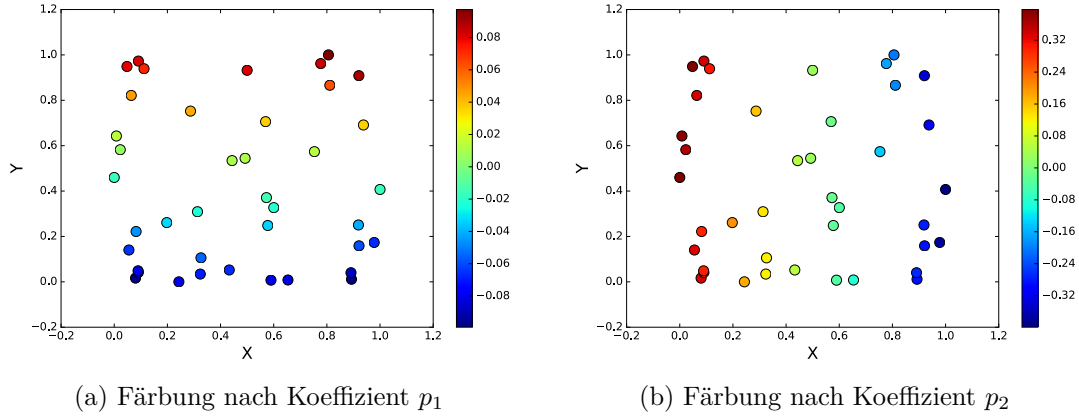
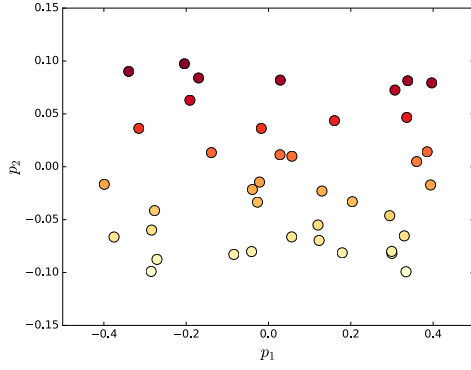
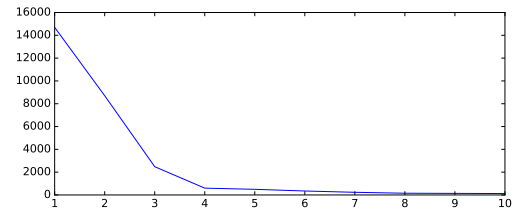


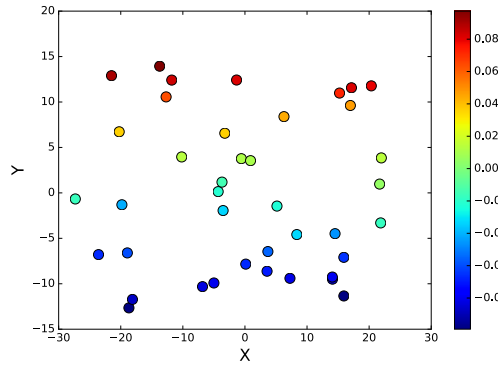
Abbildung A.1.: Einbettung $\mathcal{X}_{red}$ mit Diffusion Maps des Beispiels POLY2 in den $\mathbb{R}^2$.
Parameter: $\epsilon = 50$



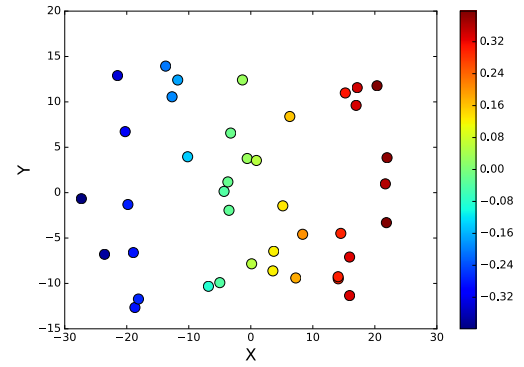
(a) Parameterraum P



(b) 10 größten Eigenwerte der Graphendistanzmatrix

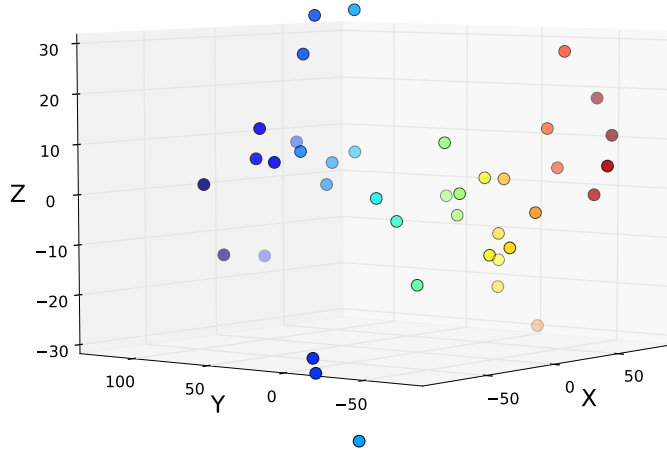


(c) Färbung nach Koeffizient p_1

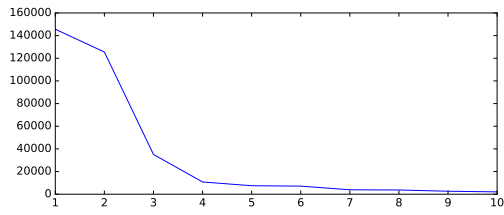


(d) Färbung nach Koeffizient p_2

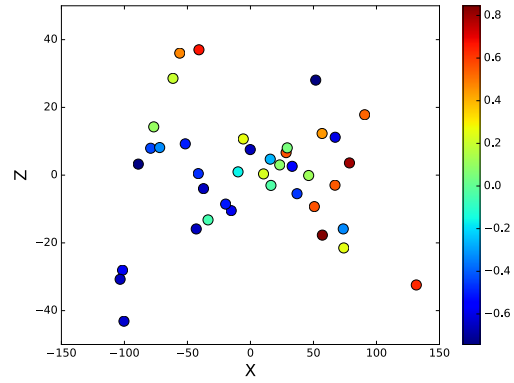
Abbildung A.2.: Einbettung $\mathcal{X}_{red}$ mit Isomap des Beispiels POLY2 in den $\mathbb{R}^2$. Nachbarschaftsgröße $k = 7$



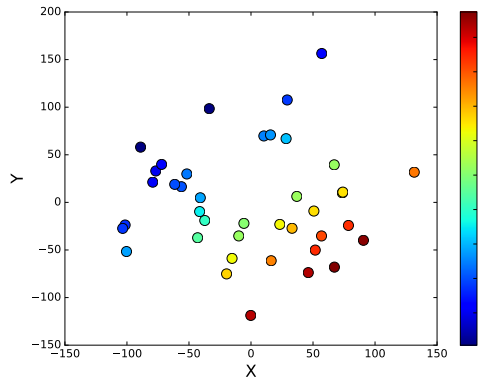
(a) Einbettung $\mathcal{X}_{red}$ im $\mathbb{R}^3$



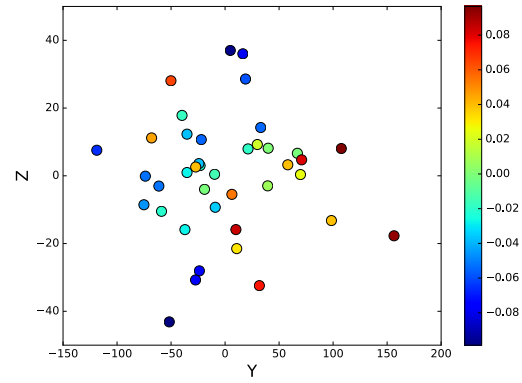
(b) 10 größten Eigenwerte der Graphendistanzmatrix



(c) X-Z-Ebene: Färbung nach Koeffizient p_1

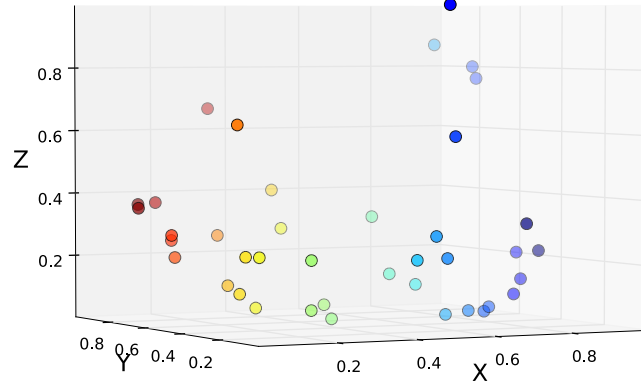


(d) X-Y-Ebene: Färbung nach Koeffizient p_2

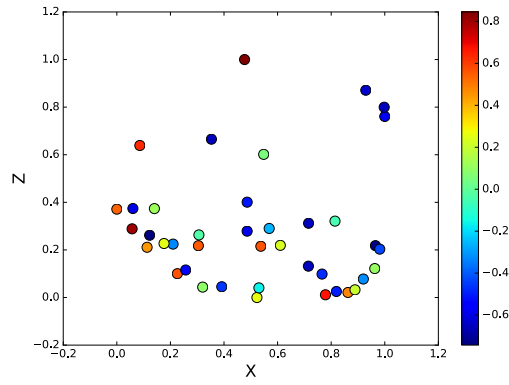


(e) Y-Z-Ebene: Färbung nach Koeffizient p_3

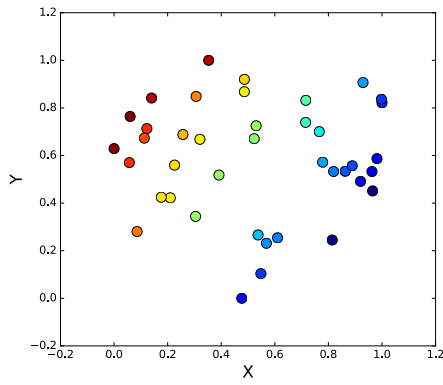
Abbildung A.3.: Einbettung mit Isomap des Beispiels POLY3 in den $\mathbb{R}^3$. Nachbarschaftsgröße $k = 7$



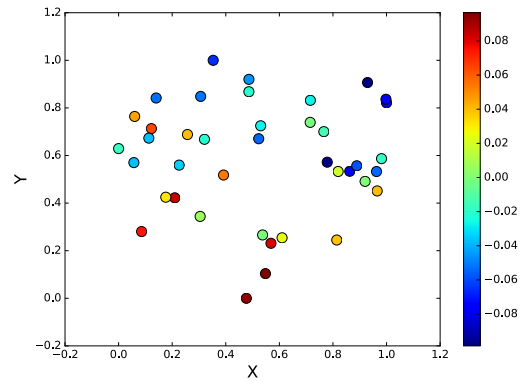
(a) Einbettung $\mathcal{X}_{red}$ im $\mathbb{R}^3$



(b) X-Z-Ebene: Färbung nach Koeffizient p_1

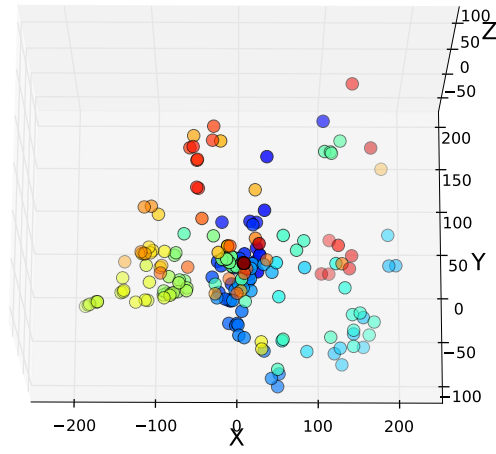


(c) X-Y-Ebene: Färbung nach Koeffizient p_2

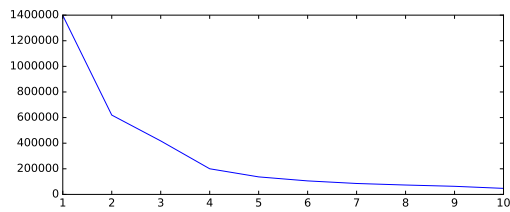


(d) X-Y-Ebene: Färbung nach Koeffizient p_3

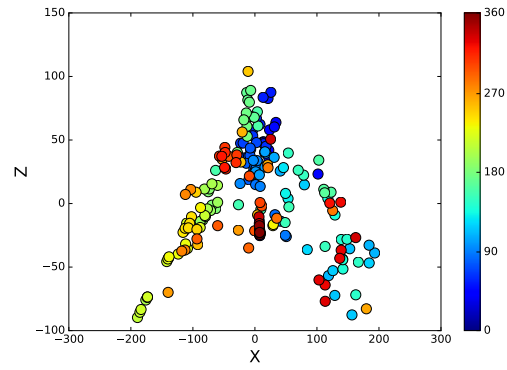
Abbildung A.4.: Einbettung mit Diffusion Maps des Beispiels POLY3 in den $\mathbb{R}^3$. Parameter: $\epsilon = 50$



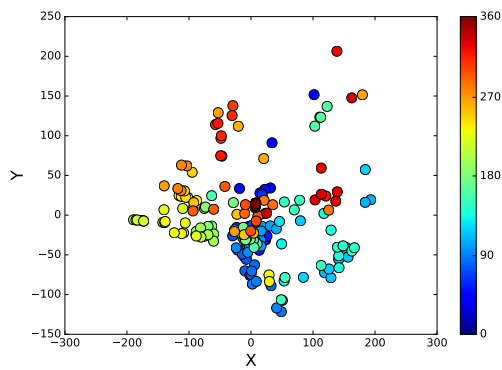
(a) Einbettung $\mathcal{X}_{red}$ im $\mathbb{R}^3$



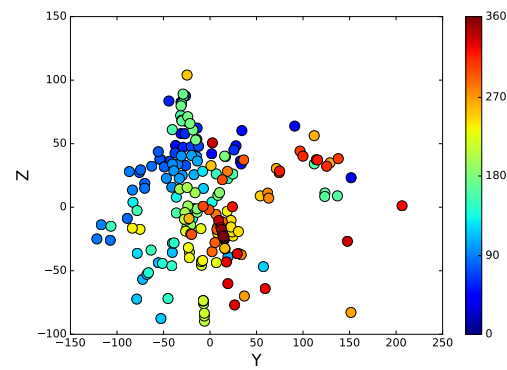
(b) 10 größten Eigenwerte der Graphendistanzmatrix



(c) X-Z-Ebene

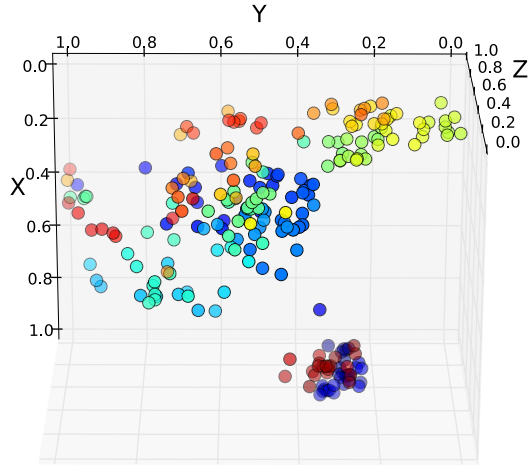


(d) X-Y-Ebene

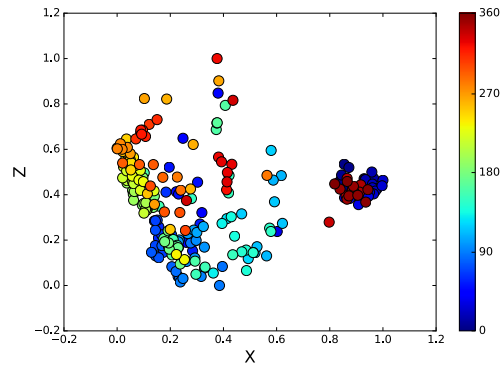


(e) Y-Z-Ebene

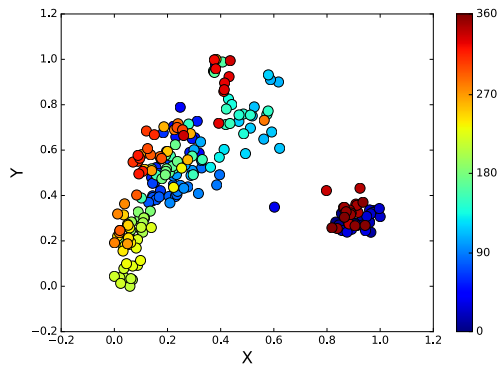
Abbildung A.5.: Einbettung von Isomap der Anwendung PKW-CRASH in den $\mathbb{R}^3$ mit Einfärbung nach Winkelposition. Nachbarschaftsgröße $k=7$



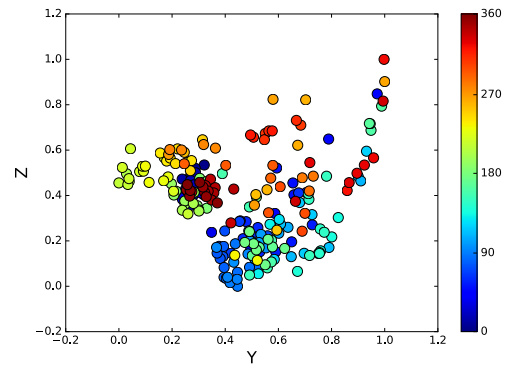
(a) Einbettung $\mathcal{X}_{red}$ im $\mathbb{R}^3$



(b) X-Z-Ebene

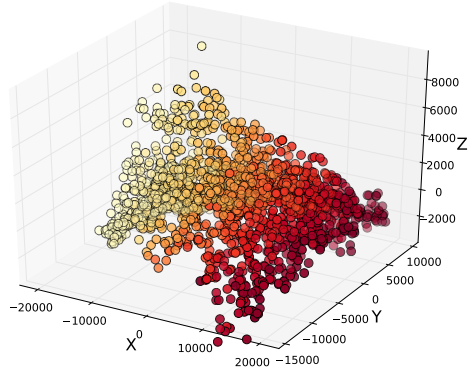


(c) X-Y-Ebene

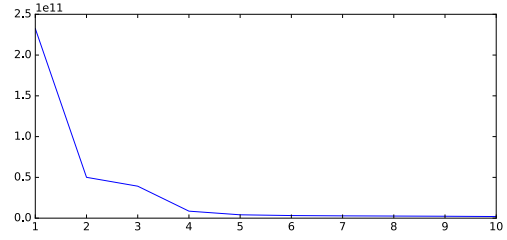


(d) Y-Z-Ebene

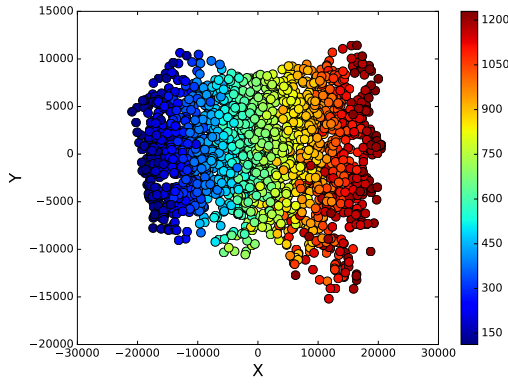
Abbildung A.6.: Einbettung von Diffusion Maps der Anwendung PKW-CRASH in den $\mathbb{R}^3$ mit Einfärbung nach Winkelposition. Parameter: $\epsilon=5000$



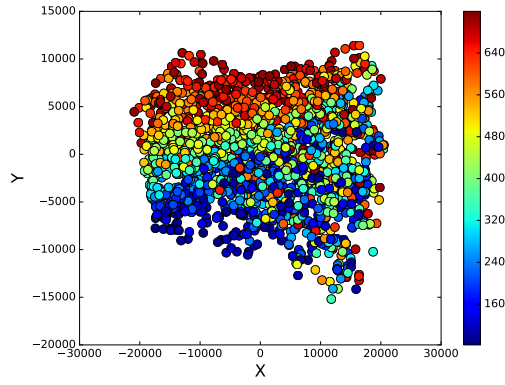
(a) Einbettung $\mathcal{X}_{red}$ im $\mathbb{R}^3$



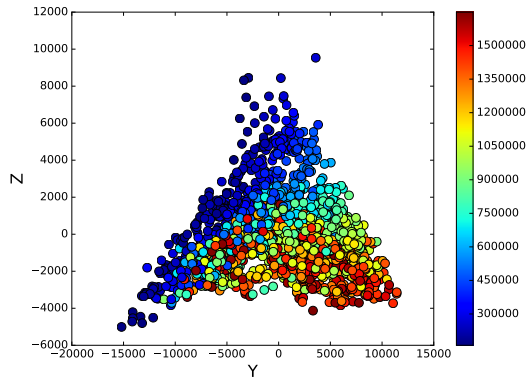
(b) 10 größten Eigenwerte der Graphendistanzmatrix



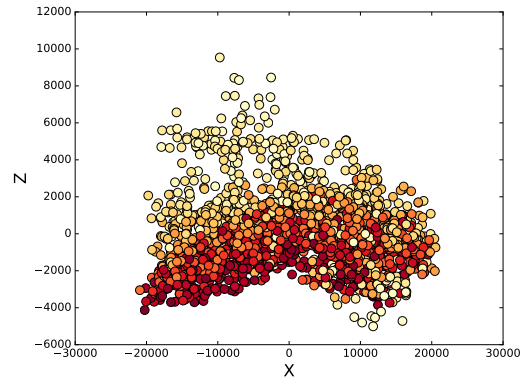
(c) X-Y-Ebene: Färbung nach Parameter **cb**



(d) X-Y-Ebene: Färbung nach Parameter **y**

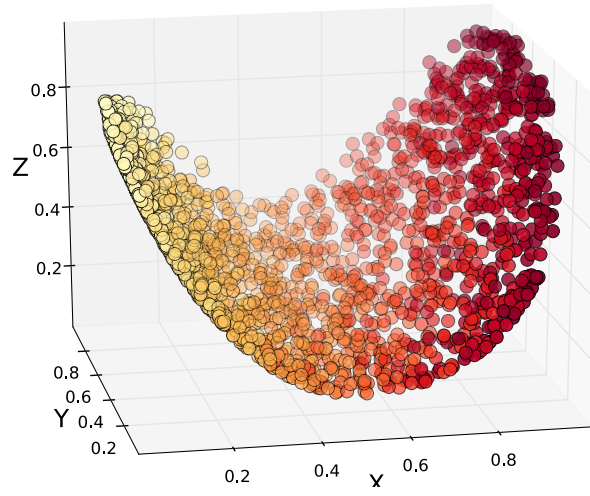


(e) Y-Z-Ebene: Färbung nach Parameter **ea**

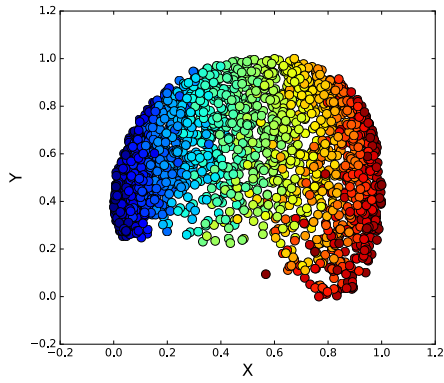


(f) X-Z-Ebene

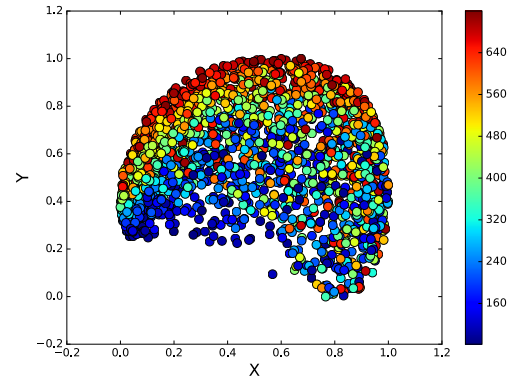
Abbildung A.7.: Einbettung von Isomap der Anwendung ZUGVERSUCH in den $\mathbb{R}^3$.
Nachbarschaftsgröße $k=7$



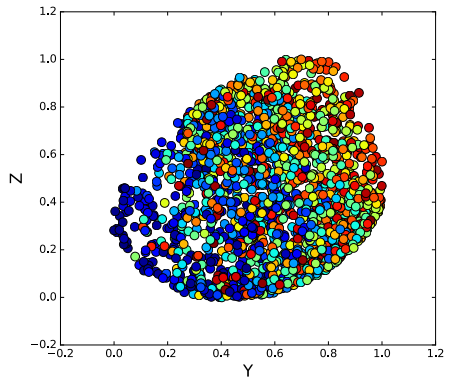
(a) Einbettung $\mathcal{X}_{red}$ im $\mathbb{R}^3$



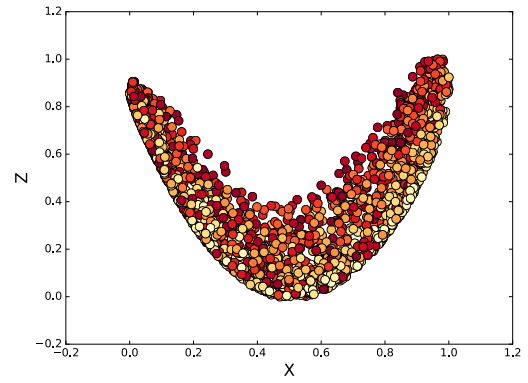
(b) X-Y-Ebene: Färbung nach Parameter **cb**



(c) X-Y-Ebene: Färbung nach Parameter **y**



(d) Y-Z-Ebene: Färbung nach Parameter **ea**



(e) X-Z-Ebene

Abbildung A.8.: Einbettung mit Diffusion Maps der Anwendung ZUGVERSUCH in den $\mathbb{R}^3$. Parameter: $\epsilon=50000$

Abbildungsverzeichnis

1.1.	Dimensionsreduktion des Beispiels ZUGVERSUCH mit Diffusion Maps. Parameter: $\epsilon = 50000$	2
1.2.	Mengen und Abbildungen bei der Rückabbildung	3
2.1.	Einbettung mit Isomap von 698 Aufnahmen einer Büste in den $\mathbb{R}^3$. Nachbarschaftsgröße $k=6$. Quelle: [TSL00]	6
2.2.	Veranschaulichung einer PCA angewandt auf dreidimensionale Daten, welche von der Mannigfaltigkeit $X=Z$ mit Rauschen gesampelt wurden .	7
2.3.	Abstandskonzepte bei Dimensionsreduktionen	10
2.4.	Nachbarbarschaftsgraphen zwei verschiedener Swissrolls. Nachbarschaftsgröße $k=5$	12
2.5.	Synthetisches Beispiel POLY2. 40 Parameter $(p_1^i, p_2^i) \in [-0.4, 0.4] \times [-0.1, 0.1]$ als Simulationseingaben, $i = 1, \dots, 40$	12
2.6.	Graphische Darstellung der einzelnen Schritte des LLE. Quelle: [RS00] .	15
2.7.	Veranschaulichung der Diffusionsmetrik. Quelle: [Laf04]	17
3.1.	Übersicht an der Rückabbildung beteiligter Funktionen und Räume . . .	19
3.2.	Einfluss der Wahl von l auf die Straftermgewichte	22
4.1.	Evaluierte Polynome des synthetischen Beispiels POLY3. 40 Parameter $(p_1^i, p_2^i, p_3^i) \in [-0.75, 0.85] \times [-0.4, 0.4] \times [-0.1, 0.1]$ als Simulationseingaben, $i = 1, \dots, 40$	28
4.2.	Ablauf der Simulationen des Beispiels ZUGVERSUCH	28
4.3.	Simulationen der Anwendung PKW-CRASH	29
4.4.	Approximationsfehler der Transformation. Parameter des Gauss-Kerns: POLY2: $\epsilon = 0.15$, POLY3: $\epsilon = 0.5$, PKW-CRASH: $\epsilon = 1.3$ und ZUGVERSUCH: $\epsilon = 500$	34
4.5.	Approximationsfehler der Inversion der Dimensionsreduktion mit LLI .	36
4.6.	Durchschnittliche Gewichte der 14 nächsten Nachbarn bei den Linearkombination und Entfernungen dieser Nachbarn in $\mathcal{X}$ für die Anwendung Zugversuch	37
4.7.	Approximationsfehler der kompletten Rückabbildung in Abhängigkeit der Nachbarschaftsgröße k mit den Parametern $\epsilon = 0.01$ und $l = 4$	39
4.8.	Fehler der Rückabbildung unter Verwendung der Nachbarn aus $\mathcal{P}$ beim Beispiel POLY3. Nachbarschaftsgröße $k = 3, 4, \dots, 14$, $\epsilon = 0.01$, $l = 4$. .	40

4.9. Vergleich der Approximationen in $\mathcal{X}_{red}$ und $\mathcal{X}$ eines durchschnittlichen Beispiels der Anwendung ZUGVERSUCH. Parameter: Diffusion Maps mit $\epsilon = 50000$ und LLI mit $k = 10$, $l = 4$ und $\epsilon = 0.01$	42
4.10. Durchschnittlicher relativer Approximationsfehler der Rückabbildung in Abhängigkeit der LLI-Nachbarschaftsgröße unter Verwendung der optimalen Einstellungen	43
A.1. Einbettung $\mathcal{X}_{red}$ mit Diffusion Maps des Beispiels POLY2 in den $\mathbb{R}^2$. Parameter: $\epsilon = 50$	49
A.2. Einbettung $\mathcal{X}_{red}$ mit Isomap des Beispiels POLY2 in den $\mathbb{R}^2$. Nachbarschaftsgröße $k = 7$	50
A.3. Einbettung mit Isomap des Beispiels POLY3 in den $\mathbb{R}^3$. Nachbarschaftsgröße $k = 7$	51
A.4. Einbettung mit Diffusion Maps des Beispiels POLY3 in den $\mathbb{R}^3$. Parameter: $\epsilon = 50$	52
A.5. Einbettung von Isomap der Anwendung PKW-CRASH in den $\mathbb{R}^3$ mit Einfärbung nach Winkelposition. Nachbarschaftsgröße $k=7$	53
A.6. Einbettung von Diffusion Maps der Anwendung PKW-CRASH in den $\mathbb{R}^3$ mit Einfärbung nach Winkelposition. Parameter: $\epsilon=5000$	54
A.7. Einbettung von Isomap der Anwendung ZUGVERSUCH in den $\mathbb{R}^3$. Nachbarschaftsgröße $k=7$	55
A.8. Einbettung mit Diffusion Maps der Anwendung ZUGVERSUCH in den $\mathbb{R}^3$. Parameter: $\epsilon=50000$	56

Literatur

- [Boh+13] B. Bohn u. a. “Analysis of Car Crash Simulation Data with Nonlinear Machine Learning Methods”. In: *Procedia Computer Science* 18 (2013). International Conference on Computational Science 2013, S. 621–630.
- [CL06] Ronald R. Coifman und Stéphane Lafon. “Diffusion maps”. In: *Appl. Comput. Harmon. Anal.* 21.1 (2006), S. 5–30.
- [Fis08] Gerd Fischer. *Lineare Algebra*. Eine Einführung für Studienanfänger. Vieweg + Teubner, Wiesbaden, 2008.
- [Fra+14] T. Franz u. a. “Interpolation-based reduced-order modelling for steady transonic flows via manifold learning”. In: *Int. J. Comput. Fluid Dyn.* 28.3-4 (2014), S. 106–121.
- [Fra15] Thomas Franz. “Reduced-order modeling for steady transonic flows via manifold learning”. Diss. Technische Universität Carolo-Wilhelmina zu Braunschweig, 2015.
- [GK02] Carl Geiger und Christian Kanzow. *Theorie und Numerik restringierter Optimierungsaufgaben*. 1. Aufl. Springer-Verlag Berlin Heidelberg, 2002.
- [Gri+14] M. Griebel u. a. *SIMDATA-NL - Nichtlineare Charakterisierung und Analyse von FEM-Simulationsergebnissen für Autobauteile und Crash-Tests*. Abschlussbericht des BMBF-Projekts, 2014. 2014.
- [Han09] Martin Hanke-Bourgeois. *Grundlagen der numerischen Mathematik und des wissenschaftlichen Rechnens*. 3. Aufl. Vieweg + Teubner, Wiesbaden, 2009, S. 840.
- [Hoa62] C. A. R. Hoare. “Quicksort”. In: *The Computer Journal* 5.1 (1962), S. 10–16.
- [IG] R. Iza-Teran und J. Garcke. *Data Analytics for Simulation Repositories in Industry*.
- [Jol86] I. T. Jolliffe. *Principal component analysis*. Springer Series in Statistics. Springer-Verlag, New York, 1986, S. xiv+271.
- [KV12] Bernhard Korte und Jens Vygen. *Kombinatorische Optimierung. Theorie und Algorithmen*. 2. Aufl. Übersetzt von Rabe von Randow. Springer-Verlag Berlin Heidelberg, 2012.
- [Laf04] Stéphane S. Lafon. *Diffusion maps and geometric harmonics*. Thesis (Ph.D.)–Yale University. ProQuest LLC, Ann Arbor, MI, 2004, S. 125.
- [LV07] John A. Lee und Michel Verleysen. *Nonlinear dimensionality reduction*. Information Science and Statistics. Springer-Verlag, New York, 2007, S. xviii+308.

- [Mei11] Andreas Meister. *Numerik linearer Gleichungssysteme*. 4. Aufl. Eine Einführung in moderne Verfahren. Vieweg + Teubner, Wiesbaden, 2011.
- [NLK06] Boaz Nadler, Ronald R. Lafon Stéphane Coifman und Ioannis G. Kevrekidis. “Diffusion maps, spectral clustering and reaction coordinates of dynamical systems”. In: *Appl. Comput. Harmon. Anal.* 21 (2006), S. 113–127.
- [Ped+11] Fabian Pedregosa u. a. “Scikit-learn: machine learning in Python”. In: *J. Mach. Learn. Res.* 12 (2011), S. 2825–2830.
- [RS00] Sam T. Roweis und Lawrence K. Saul. “Nonlinear Dimensionality Reduction by Locally Linear Embedding”. In: 290(5500) (2000), S. 2323–2326.
- [SW06] Robert Schaback und Holger Wendland. “Kernel techniques: from machine learning to meshless methods”. In: *Acta Numer.* 15 (2006), S. 543–639.
- [TSK09] Nicolas Thorstensen, Florent Segonne und Renaud Keriven. “Pre-image as Karcher Mean Using Diffusion Maps: Application to Shape and Image Denoising”. In: *Scale Space and Variational Methods in Computer Vision. SSVM 2009*. Hrsg. von Xue-Cheng Tai u. a. Springer-Verlag Berlin Heidelberg, 2009, S. 721–732.
- [TSL00] Joshua B. Tenenbaum, Vin de Silva und John C. Langford. “A Global Geometric Framework for Nonlinear Dimensionality Reduction”. In: 290(5500) (2000), S. 2319–2323.
- [UU12] Michael Ulbrich und Stefan Ulbrich. *Nichtlineare Optimierung*. 1. Aufl. Birkhäuser Basel, 2012.