Carl von Ossietzky Universität Oldenburg
Fakultät II – Informatik, Wirtschafts- und Rechtswissenschaften
Studiengang: Master Informatik

Masterarbeit

Ein experimenteller Vergleich von Dimensionsreduktionsverfahren

vorgelegt von Almuth Meier

Oldenburg, 14. Dezember 2016

Betreuender Gutachter: Jun.-Prof. Dr. habil. Oliver Kramer
Zweiter Gutachter: Daniel Lückehe, M.Sc., B.Eng.

Inhaltsverzeichnis

1 Einführung **1**
1.1 Bedeutung der Dimensionsreduktion 1
1.2 Bisherige Vergleiche von Dimensionsreduktionsverfahren . . . 3
1.3 Ziel der Arbeit 4
1.4 Aufbau der Arbeit 5

2 Dimensionsreduktionsverfahren **7**
2.1 Informationserhaltung 9
2.2 Ordnungsschema für Dimensionsreduktionsverfahren 11
2.3 Konvexe Dimensionsreduktionsverfahren 14
2.3.1 Vollspektrale Verfahren 15
2.3.2 Teilspektrale Verfahren 27
2.4 Nichtkonvexe Dimensionsreduktionsverfahren 34
2.4.1 Multidimensional Scaling (MDS) 34
2.4.2 Probabilistische Verfahren 37
2.4.3 Unüberwachte Regressionsverfahren 47
2.4.4 Künstliche neuronale Netze 52
2.5 Erweiterungen von Dimensionsreduktionsverfahren 53
2.6 Eigenschaften der Dimensionsreduktionsverfahren 55

3 Experimenteller Vergleich der Dimensionsreduktionsverfahren **61**
3.1 Evaluationskonzept 61
3.1.1 Metriken 61
3.1.2 Datensätze 69
3.1.3 Verwendete Implementierungen 72
3.1.4 Einstellung der Parameter 73

3.2 Vergleich der konvexen Verfahren 74
3.2.1 Vergleich hinsichtlich der Eigenschaften 76
3.2.2 Vergleich hinsichtlich der Datensätze 79
3.2.3 Vergleich der Visualisierungen 82
3.2.4 Zusammenfassung 89
3.3 Vergleich der nichtkonvexen Verfahren 90
3.3.1 Vergleich hinsichtlich der Eigenschaften 90
3.3.2 Vergleich hinsichtlich der Datensätze 94
3.3.3 Vergleich der Visualisierungen 95
3.3.4 Zusammenfassung 103
3.4 Gemeinsamer Vergleich 103
3.4.1 Vergleich hinsichtlich der Eigenschaften und Datensätze 104
3.4.2 Vergleich des Anteils am minimalen Fehler (AMF) . . 105
3.4.3 Statistische Signifikanz der Metrikwerte 108
3.4.4 Zusammenfassung 109

4 Schluss 111
4.1 Empfehlungen zur Wahl von Dimensionsreduktionsverfahren . 111
4.2 Erweiterungsmöglichkeiten des Vergleichs 113
4.3 Zusammenfassung 114

A Wertebereiche der Parameter 117

B Inhalt der beigelegten DVD 119

Literaturverzeichnis 121

Index 137

Selbstständigkeitserklärung 141

Abbildungsverzeichnis

2.1 Swiss-roll 10
2.2 Ordnungsschema für Dimensionsreduktionsverfahren 15
2.3 Euklidische vs. geodätische Distanz 21

3.1 Künstliche Datensätze 72
3.2 Durchschnittliche Ränge konvexer Verfahren pro Metrik . . . 76
3.3 Häufigkeiten für Ränge ≤ 5 bzw. ≥ 20 konvexer Verfahren . . 79
3.4 Durchschnittliche Ränge konvexer Verfahren pro Datensatz . 80
3.5 Durchschnittliche Ränge nichtkonvexer Verfahren pro Metrik 90
3.6 Häufigkeiten für Ränge ≤ 5 bzw. ≥ 20 nichtkonvexer Verfahren 92
3.7 Durchschnittliche Ränge nichtkonvexer Verfahren pro Datensatz 93
3.8 Durchschnittliche Ränge der besten konvexen und nichtkonvexen Verfahren pro Metrik 104
3.9 Durchschnittliche Ränge der besten konvexen und nichtkonvexen Verfahren pro Datensatz 105
3.10 AMF für konvexe und nichtkonvexe Verfahren 106

Tabellenverzeichnis

2.1 Eigenschaften der konvexen Dimensionsreduktionsverfahren . 56
2.2 Eigenschaften der nichtkonvexen Dimensionsreduktionsverfahren 57

3.1 Eigenschaften der realen Datensätze 70
3.2 Visualisierte Einbettungen konvexer Verfahren (künstliche Datensätze) 86
3.3 Visualisierte Einbettungen konvexer Verfahren (reale Datensätze) 89
3.4 Visualisierte Einbettungen nichtkonvexer Verfahren (künstliche Datensätze) 99
3.5 Visualisierte Einbettungen nichtkonvexer Verfahren (reale Datensätze) 103
3.6 p-Werte der Friedman-Tests 109

A.1 Wertebereiche der Parameter der konvexen Verfahren 117
A.2 Wertebereiche der Parameter der nichtkonvexen Verfahren . . 118

Abkürzungsverzeichnis

AMF Anteil am minimalen Fehler

ANN künstliches neuronales Netz

CE Conformal Eigenmaps

CRR Classification Rate Reduction

CS Classical Scaling

DM Diffusion Maps

DSRE Datenraumrekonstruktionsfehler

E1NN 1NN-Fehler

EE Elastic Embedding

EKNN KNN-Fehler

EKS Kruskal-Shepard-Fehler

ENX Nachbarschafts-Fehler

FA Factor Analysis

GPLVM Gaussian Process Latent Variable Model

HLLE Hessian Locally Linear Embedding

Isomap Isometric Feature Mapping

itUKR iterative UKR

KNN k-nächste-Nachbarn

KPCA Kernel PCA

LE Laplacian Eigenmaps

LLC Locally Linear Coordination

LLE Locally Linear Embedding

LLTSA Linear LTSA

LPP Locality Preserving Projections

LTSA Local Tangent Space Alignment

MC Manifold Charting

MDS Multidimensional Scaling

MFA mixture of factor analyzers

MLLE Modified Locally Linear Embedding

MM mixture model

mMDS metric MDS

MVU Maximum Variance Unfolding

nMDS nonmetric MDS

NPE Neighborhood Preserving Embedding

NPP Neighborhood Preserving Projections

PCA Principal Components Analysis

PPCA Probabilistic PCA

SM Sammon Mapping

SNE Stochastic Neighbor Embedding

SOM self-organizing map

t-SNE t-Distributed SNE

UKR Unsupervised Kernel Regression

UNN Unsupervised Nearest Neighbors

Symbolverzeichnis

$\mathbf{E}$	Einheitsmatrix
$\mathbf{K}$	Kern-/Gram-Matrix
$\mathbf{L}$	Matrix der latenten Punkte, $\mathbf{L} \in \mathbb{R}^{n \times q}$
$\mathbf{l}$	latenter Punkt
$\mathbf{n}$	Anzahl Muster des Datensatzes
$\mathbf{P}$	Matrix der Muster, $\mathbf{P} \in \mathbb{R}^{n \times d}$
$\mathbf{p}$	Muster
$\mathbf{X}$	Matrix
$\mathbf{x}$	Vektor
$\mathbf{X}^T$	transponierte Matrix
$\mathbf{x}_i$	i-ter Vektor
$\|\mathbf{x}\|$	euklidische Norm, $\|\mathbf{x}\| = \sqrt{\sum_i x_i^2}$
$\|\mathbf{x}\|^2$	quadrierte euklidische Norm
$\|\mathbf{X}\|_F$	Frobenius-Norm, $\|\mathbf{X}\|_F = \sqrt{\sum_i \sum_j x_{ij}^2}$
$\|\mathbf{X}\|_F^2$	quadrierte Frobenius-Norm

d	Dimensionalität des Datenraums
G	Nachbarschaftsgraph
K	Kern/Kernfunktion
k	Nachbarschaftsgröße
q	Dimensionalität des latenten Raums
x_i	Eintrag des Vektors $\mathbf{x}$
x_{ij}	Eintrag der Matrix $\mathbf{X}$

1 Einführung

1.1 Bedeutung der Dimensionsreduktion

In immer mehr Forschungs- und Unternehmensbereichen werden umfangreiche Daten erhoben, analysiert und gespeichert, um beispielsweise erfolgreiche Unternehmensstrategien zu entwickeln oder durch Simulationen neue Erkenntnisse über einen Untersuchungsgegenstand zu gewinnen. Dabei stellt nicht nur die Menge der Daten, sondern auch deren Komplexität eine Herausforderung bei der Analyse der Daten dar. Ist für den Menschen eine dreidimensionale Grafik zur Visualisierung von Daten noch akzeptabel, übersteigen bereits vierdimensionale Daten das menschliche Vorstellungsvermögen, insbesondere da eine grafische Darstellung nur schwer möglich ist. Dieser Umstand ist problematisch, da beispielsweise selbst einfache Kundendaten eines Unternehmens schon vierdimensional sind, wenn pro Kunde die Merkmale Name, Adresse, Geburtsdatum und bestellte Ware erhoben werden. Im Vergleich zu vielen anderen Daten sind diese Kundendaten allerdings noch wenig komplex: Im Gesundheitsbereich können bei der Erforschung von Krankheiten oder des menschlichen Genoms Daten mit mehreren hundert oder über tausend Dimensionen entstehen. Bei der Bilderkennung und -verarbeitung können ebenfalls Daten mit mehreren hundert Dimensionen anfallen [WMS00]. Auch im Bereich der Energieversorgung werden hoch-dimensionale Daten eingesetzt. Um beispielsweise die von einem Windpark erzeugte Leistung zu untersuchen, werden teilweise Datensätze verwendet, die für jede Anlage im Windpark deren erzeugte Leistung enthalten. Hier entspricht die Dimensionalität der Daten

der Anlagenanzahl [Kra13].

Bei der Datenanalyse ist eine hohe Dimensionalität nicht nur für den Menschen eine Herausforderung, sondern auch beim Einsatz maschineller Lernverfahren. Zum Lernen benötigen die Algorithmen Trainingsdaten, welche die in der Wirklichkeit beobachtbaren Daten und deren Verteilung (d. h. den *Datenraum*) gut abdecken. Umfassen die Trainingsdaten einen bestimmten Teil des Datenraums nicht, erzielt der Algorithmus im praktischen Einsatz für Daten aus diesem Bereich schlechte Lernergebnisse. Mit zunehmender Dimensionalität steigt die Anzahl an Trainingsdaten, die vorliegen muss, um eine gleichbleibend gute Abdeckung des Datenraums zu gewährleisten, exponentiell (*Fluch der Dimensionen*).[1] Dies ist nachteilig, da sich das Training der meisten Algorithmen sowohl mit steigender Dimensionalität der Daten als auch mit wachsendem Umfang des Trainingsdatensatzes verlangsamt [Mar09].

Aus diesen Gründen wurden Dimensionsreduktionsverfahren entwickelt, welche die Dimensionalität (d. h. die Anzahl an Merkmalen) der beobachteten *Muster* (d. h. der erhobenen Daten) verringern. In der Literatur wurden verschiedene Ansätze vorgestellt, die von *Merkmalsauswahl* über *Gruppierung* anhand ähnlicher Merkmale bis hin zur Berechnung neuer Merkmale aus den ursprünglichen Merkmalen (*Merkmalsextraktion*) reichen [Mar09; Kra13; Alp10]. Innerhalb dieser Gruppen existiert jeweils eine Vielzahl von Verfahren aus verschiedenen *Familien*, die sich in der Herangehensweise und den verwendeten mathematischen Methoden unterscheiden.

Diese Vielfalt an Dimensionsreduktionsverfahren erschwert in der Praxis die Auswahl des richtigen Verfahrens für den konkreten Anwendungsfall, da neu vorgestellte Verfahren oftmals nur gegenüber den bisherigen Standardverfahren nicht aber gegenüber anderen neuartigen Verfahren evaluiert werden. In der Literatur existieren einige Arbeiten, welche die Ergebnisse verschiedener Verfahren bei der Reduktion unterschiedlicher Datensätze anhand

[1] Der Begriff wurde laut [HTF09] von [Bel61] geprägt.

quantitativer Qualitätsmaße (*Metriken*) vergleichen und Hilfestellung bei der Auswahl geben können. Diese werden im nächsten Abschnitt vorgestellt.

1.2 Bisherige Vergleiche von Dimensionsreduktionsverfahren

Die umfangreichste Vergleichsstudie ist [MPH09] mit 13 Dimensionsreduktionsverfahren, 10 Datensätzen und drei Metriken. Neuere Verfahren wie Unsupervised Nearest Neighbors [Kra13], Elastic Embedding [CP10] oder t-SNE [MH08] werden jedoch nicht berücksichtigt. In [GH15] werden zwar mehr Dimensionsreduktionsverfahren miteinander verglichen (18), jedoch wird nur eine Metrik verwendet; diese Arbeit legt den Schwerpunkt auf den Vergleich der visualisierten Ergebnisse. Dagegen ist [MHP11] weniger auf den Vergleich unterschiedlicher Dimensionsreduktionsverfahren als vielmehr auf die Untersuchung des Verhaltens der Verfahren bei unterschiedlichen Eigenschaften der Datensätze ausgerichtet. Es werden fünf künstliche Datensätze eingesetzt, auf denen die vier evaluierten Verfahren wiederholt ausgeführt werden. Bei jeder Wiederholung werden die Eigenschaften der Datensätze wie zum Beispiel die Dichte oder die Stärke des Rauschens variiert. Die Ergebnisse werden anhand zweier Metriken bewertet. Neben diesen Arbeiten existieren Vergleiche, welche die Verwendbarkeit von wenigen ausgewählten Dimensionsreduktionsverfahren für einen speziellen Anwendungszweck untersuchen, zum Beispiel [Hal+12] (Texte aus dem Gesundheitswesen), [BAM12] (Speicherung von Sonnenbildern) oder [NS03] (visuelle Kontrolle von Holzoberflächen).

Insgesamt sind diese Arbeiten jedoch für eine begründete Auswahl eines Verfahrens nicht umfassend genug, da entweder zu wenig Dimensionsreduktionsverfahren in den Vergleich einbezogen oder nicht ausreichend Datensätze oder Metriken zur Bewertung der Verfahren verwendet werden. Somit ist keine Übersicht über praktisch ermittelte und quantifizierte Eigenschaften

von Dimensionsreduktionsverfahren vorhanden, welche eine gezielte Auswahl von Dimensionsreduktionsverfahren zum Einsatz in der Forschung oder im Unternehmen ermöglicht.

1.3 Ziel der Arbeit

Diese Arbeit hat das Ziel, eine solche Übersicht auf der Grundlage eines umfangreichen experimentellen Vergleichs zu erstellen. Dazu wird ein Konzept entworfen, das festlegt, anhand welcher Metriken und Datensätze Dimensionsreduktionsverfahren sinnvoll bewertet und miteinander verglichen werden können. Dieses Konzept wird praktisch umgesetzt, indem eine ausgewählte Menge von Dimensionsreduktionsverfahren für die festgelegten Datensätze ausgeführt wird und die Resultate durch die Metriken bewertet werden. Hierfür wird ein Rahmengerüst implementiert, das die Ausführung der Dimensionsreduktionsverfahren anstößt und die Ergebnisse der Metriken in geeigneter Form speichert.

Um diese Arbeit von bereits vorgestellten Vergleichsstudien abzugrenzen, werden Dimensionsreduktionsverfahren aus möglichst vielen verschiedenen Familien und auch neuere Verfahren in den experimentellen Vergleich aufgenommen. Bei der Auswahl der Datensätze wird darauf geachtet, dass diese verschiedenartig sind, sich also beispielsweise merklich in der Anzahl an Dimensionen unterscheiden, um die Stärken und Schwächen der Verfahren deutlich erkennen zu können. Die Metriken werden so zusammengestellt, dass unterschiedliche Eigenschaften der Algorithmen gemessen werden. Mit dieser Kombination von Datensätzen und Metriken wird die Wahrscheinlichkeit reduziert, die Stärken beziehungsweise Schwächen eines Verfahrens in einem besonderen Anwendungsfall aufgrund eines zu allgemeinen Evaluationsansatzes zu übersehen. Insgesamt werden 29 Dimensionsreduktionsverfahren aus unterschiedlichen Familien auf 13 künstlichen und realen Datensätzen ausgeführt und mit Hilfe von sechs Metriken bewertet.

1.4 Aufbau der Arbeit

Die vorliegende Arbeit ist in zwei Hauptteile gegliedert. Im ersten Teil (Kapitel 2) wird in das Themenfeld der Dimensionsreduktion eingeführt, zudem werden die im experimentellen Vergleich verwendeten Dimensionsreduktionsverfahren vorgestellt. Im zweiten Teil (Kapitel 3) werden das Konzept der Evaluation erläutert und die Ergebnisse der Experimente ausgewertet. Abschließend (Kapitel 4) werden hieraus Empfehlungen zur Wahl von Dimensionsreduktionsverfahren in der Praxis abgeleitet, Erweiterungsmöglichkeiten dieser Arbeit aufgezeigt und die Ergebnisse der Arbeit zusammengefasst.

2 Dimensionsreduktionsverfahren

Als Dimensionsreduktion wird das Verringern der Dimensionalität d eines hoch-dimensionalen Datensatzes auf q Dimensionen verstanden. Dies bedeutet, dass die Datenobjekte (Muster) des Datensatzes nach der Dimensionsreduktion nur mit q anstatt mit d Merkmalen beschrieben werden. Die niedrig-dimensionalen Äquivalente der Muster werden als *latente Punkte* oder *latente Einbettungen*[1] bezeichnet. Der Raum, aus dem die Muster stammen, ist der *Datenraum* und ist häufig der $\mathbb{R}^d$. Der Raum der latenten Punkte, wird als *latenter Raum* bezeichnet und ist analog oft der $\mathbb{R}^q$, wobei $q < d$. Zur Formalisierung der Dimensionsreduktionsverfahren werden die Datensätze in Form von Matrizen geschrieben: $\mathbf{P} \in \mathbb{R}^{n \times d}$ als Matrix für die Muster und $\mathbf{L} \in \mathbb{R}^{n \times q}$ als Matrix für die latenten Punkte, wobei n die Anzahl an Mustern bzw. latenten Punkten angibt. Ein einzelnes Muster wird als $\mathbf{p}_i \in \mathbb{R}^d$ bezeichnet und entspricht einer Zeile in $\mathbf{P}$. Analog ist $\mathbf{l}_i \in \mathbb{R}^q$ ein latenter Punkt, der als Zeile in $\mathbf{L}$ enthalten ist. Formal ausgedrückt ist Dimensionsreduktion eine Abbildung $\mathbf{F} : \mathbb{R}^d \to \mathbb{R}^q$ vom Daten- in den latenten Raum.

Die Umsetzung dieser Abbildung variiert von Verfahren zu Verfahren. Wie schon in Kapitel 1 erwähnt, existieren grundsätzlich drei Gruppen dimensionsreduzierender Verfahren. Bei der Merkmalsauswahl (engl. *feature selection*)

[1] In der Literatur werden die latenten Punkte teilweise auch als *latente Variablen* bezeichnet. Dieser Begriff ist bei einem speziellen Typ von Dimensionsreduktionsverfahren, den *latent variable models* (siehe Abschnitt 2.4.2), jedoch mit einer anderen Bedeutung belegt: Dort sind die Dimensionen des niedriger-dimensionalen Raums gemeint. Aus diesem Grund werden in dieser Arbeit latente Punkte nicht als latente Variablen bezeichnet.

werden von den ursprünglich d Dimensionen q zur Beschreibung der Muster übernommen, sodass möglichst viele wichtige Informationen erhalten bleiben. Bei der Gruppierung (engl. *clustering*) werden die Muster anhand ihrer Ähnlichkeit bezüglich der einzelnen Merkmale in Gruppen eingeteilt, wobei ähnliche Muster gleichen Gruppen zugewiesen werden. So wird eine Reduktion auf genau eine Dimension mit diskretem Wertebereich vorgenommen: die Gruppenzugehörigkeit. Die dritte Gruppe bilden Verfahren zur Merkmalsextraktion (engl. *feature extraction*). Diese erzeugen, im Gegensatz zur Merkmalsauswahl, neue Dimensionen. Diese werden durch Transformation der ursprünglichen Merkmale berechnet [Ize13]. Zusätzlich zu diesen drei Gruppen existieren Verfahren, die wie bei der Merkmalsextraktion neue Dimensionen generieren. Diese sind jedoch keine Kombinationen der ursprünglichen Merkmale, sondern werden auf andere Weise berechnet.

In dieser Arbeit werden ausschließlich Verfahren der beiden letztgenannten Gruppen berücksichtigt, da die Verfahren der anderen beiden Gruppen als Dimensionsreduktionsverfahren teilweise ungeeignet sind. Die Merkmalsauswahl betrachtet gegenüber der Merkmalsextraktion nur einzelne Merkmale isoliert, sodass die viel wichtigeren Informationen, die durch Kombination von Merkmalen beschrieben werden, verloren gehen [Alp10]. Gruppierungsverfahren können nur eingeschränkt als Dimensionsreduktionsverfahren eingesetzt werden, da bei ihnen die latente Dimensionalität auf eins festgelegt ist.

Im weiteren Verlauf dieses Kapitels wird zunächst erläutert, wie Dimensionsreduktionsverfahren prinzipiell vorgehen, um gute Einbettungen zu erzeugen (Abschnitt 2.1). In Abschnitt 2.2 werden grundlegende Eigenschaften von Dimensionsreduktionsverfahren beschrieben und ein Ordungsschema für Dimensionsreduktionsverfahren vorgestellt, auf dessen Grundlage die Verfahren für den experimentellen Vergleich ausgewählt wurden. Die Verfahren werden in den Abschnitten 2.3 und 2.4 beschrieben. In Abschnitt 2.5 wird auf Erweiterungsmöglichkeiten von Dimensionsreduktionsverfahren eingegangen, die theoretischen Eigenschaften der Verfahren werden in Abschnitt 2.6 zu-

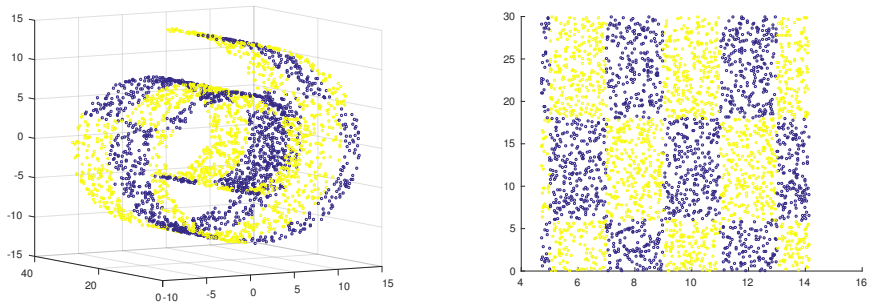
sammengefasst.

2.1 Informationserhaltung

Eine Dimensionsreduktion ist immer mit einem Informationsverlust verbunden. Deshalb werden Dimensionsreduktionsverfahren so entworfen, dass trotzdem möglichst viel wichtige Information aus dem Datenraum in den latenten Raum übernommen wird. Was dies für Information ist und wie deren Erhaltung gewährleistet werden kann, ist nicht allgemeingültig definierbar, da dies sowohl vom Datensatz als auch vom konkreten Anwendungsfall abhängt, für welchen die Dimensionsreduktion ausgeführt wird. Daher versuchen Dimensionsreduktionsverfahren, allgemeinere, topologische Informationen des Datenraums zu bewahren, um auf diesem Weg die unbekannten wichtigen Informationen in den latenten Raum zu übernehmen. Dies wird im Folgenden anhand eines Beispiels verdeutlicht.

Im linken Teil der Abbildung 2.1[2] sind die Muster des dreidimensionalen Datensatzes Swiss-roll zu sehen; sie bilden die Struktur einer Rolle. Wird diese Rolle auseinander gerollt, was einer Dimensionsreduktion entspricht, ergibt sich die im rechten Teil dargestellte Struktur. Dies ist die sogenannte *intrinsische Struktur* (auch *Mannigfaltigkeit*, engl. *manifold*) des Datensatzes. Sie weist im Wesentlichen weiterhin dieselben Eigenschaften wie die linke Struktur auf: Schachbrettmuster, rechteckige Struktur. Das Auseinanderrollen hat folglich keinen großen Informationsverlust bewirkt; nur die Spiralstruktur ist nicht mehr zu erkennen. Intuitiv erscheinen die anderen Eigenschaften (Schachbrettmuster, Rechteckstruktur) wichtiger, sodass, wenn der Anwendungskontext eine Dimensionsreduktion erfordern würde, die Spiralstruktur eine Eigenschaft wäre, auf die verzichtet werden könnte (diese Auffassung wird auch von [LV07] vertreten).

[2] Diese Abbildung wurde mit Hilfe der „Matlab Toolbox for Dimensionality Reduction“ erstellt [Maa].



(a) ursprüngliche Struktur

(b) intrinsische Struktur

Abb. 2.1: Swiss-roll

Diese Beobachtung führt zu der Frage zurück, wie Dimensionsreduktionsverfahren ihnen unbekannte aber dennoch wichtige Information bewahren können. Bei Swiss-roll sind das Schachbrettmuster und die Rechteckstruktur die unbekannten wichtigen Informationen. Das Schachbrettmuster wäre zerstört worden, wenn beim Auseinanderrollen beispielsweise Muster verschiedener Farbe ihre Positionen getauscht hätten. Es kann beispielsweise dadurch erhalten werden, dass die latenten Punkte benachbarter Muster ebenfalls benachbart angeordnet werden. Die Bewahrung der Rechteckstruktur, d. h. der unbekannten, intrinsischen Struktur, ist schwieriger sicherzustellen. Ein Dimensionsreduktionsverfahren sollte die latenten Punkte auf der Mannigfaltigkeit platzieren (*einbetten*) und diese, im Falle von Swiss-roll, beispielsweise nicht zu einem Parallelogramm verziehen oder in Stücke zerreißen. Eine Möglichkeit bietet die Erhaltung der Distanzen zwischen den Mustern. In den Abbildungen 2.1a und 2.1b ist zu sehen, dass in lokalen Bereichen die Abstände zwischen den latenten Punkten denen zwischen den zugehörigen Mustern entsprechen; es gibt keine größeren Löcher oder Häufungen latenter Punkte. Außerdem sind die latenten Punkte der am weitesten voneinander entfernten Muster (d. h. die Muster am Anfang und Ende der Rolle) auf der

Mannigfaltigkeit ebenfalls so weit wie möglich voneinander entfernt (d. h. die latenten Punkte am Anfang und Ende des Rechtecks). So können Dimensionsreduktionsverfahren durch Bewahrung topologischer Eigenschaften wie Nachbarschaften oder Distanzen die Einbettung der unbekannten Mannigfaltigkeit annähern.

Bei realen Datensätzen besteht zusätzlich das Problem, dass nicht nur die intrinsische Struktur, sondern auch die *intrinsische Dimensionalität* (die Dimensionalität der Mannigfaltigkeit) unbekannt ist. Diese muss vor der Dimensionsreduktion mittels geeigneter Verfahren geschätzt werden [MPH09; LV07], wie zum Beispiel mit dem *Maximum-Likelihood-Schätzer* [LB05]. Falls die Dimensionsreduktion zur Visualisierung eines Datensatzes eingesetzt wird, wird die intrinsische Dimensionalität stattdessen auf zwei oder drei Dimensionen gesetzt

2.2 Ordnungsschema für Dimensionsreduktionsverfahren

Da keine allgemeingültigen Kriterien für eine gute Mannigfaltigkeit beziehungsweise die wichtigsten zu bewahrenden topologischen Informationen existieren, wurde eine Vielzahl von Dimensionsreduktionsverfahren entwickelt, die jeweils unterschiedliche Strategien verfolgen, um die Muster optimal in den latenten Raum einzubetten. Dies wiederum erschwert die Einordnung der Verfahren in ein einheitliches Schema; bisher hat sich in der Literatur noch kein Schema als Referenz durchgesetzt. Ein solches ist jedoch für die Auswahl der Verfahren für den experimentellen Vergleich dieser Arbeit notwendig, damit die gewählten Verfahren das gesamte Spektrum der Dimensionsreduktionsverfahren repräsentieren. Aus diesem Grund wird für diese Arbeit ein bereits vorhandenes Schema verwendet und erweitert. Es wird am Ende dieses Abschnitts vorgestellt, nachdem zunächst Eigenschaften zur Kategorisierung von

Dimensionsreduktionsverfahren beschrieben werden.

Die wichtigsten Eigenschaften von Dimensionsreduktionsverfahren sind die folgenden (weitere sind bspw. [LV07] aufgeführt):

überwacht – unüberwacht: Überwachte Dimensionsreduktionsverfahren benötigen Muster mit Labels und nutzen diese, um die optimale Platzierung der latenten Punkte zu bestimmen. Unüberwachte Verfahren nutzen keine Labels.

linear – nichtlinear: Ein Dimensionsreduktionsverfahren wird als linear bezeichnet, wenn die Abbildungsfunktion vom Datenraum in den latenten Raum linear ist [GH15]. Die Dimensionen des latenten Raums sind somit Linearkombinationen der Dimensionen des Datenraums [LV07]. Nichtlineare Verfahren besitzen eine nichtlineare Abbildungsfunktion.

Verfahren, deren Abbildungsfunktion auf einer Matrixmultiplikation basiert und folglich linear ist, werden auch als *Projektionsmethoden* bezeichnet [EHH12]. Dieser Begriff ist jedoch nicht eindeutig definiert, denn in [Ize13] werden alle zur Verfahren zur Merkmalsextraktion und in [DKŽ13] sogar alle Dimensionsreduktionsverfahren als Projektionsmethoden bezeichnet. Nichtlineare Verfahren werden teilweise auch als *manifold learners* bezeichnet [LV07; EHH12; GH15; TKR08]. Allerdings wird auch dieser Begriff in der Literatur nicht konsistent eingesetzt. Ein Beispiel dafür ist die Definition in [MPH09], die nichtlineare Verfahren nicht als manifold learners bezeichnet, wenn sie nichtkonvex sind.Wegen ihrer Vieldeutigkeit, wird keine der beiden Bezeichnungen in dieser Arbeit verwendet.

parametrisch – nichtparametrisch: Parametrische Verfahren lernen die Abbildungfunktion vom Daten- in den latenten Raum (oder umgekehrt) und deren Parameter explizit. Nichtparametrische Verfahren lernen keine solche Funktion, sondern bilden die Muster direkt auf latente Punkte

ab [MPH09; GH15].

konvex – nichtkonvex: Dimensionsreduktionsverfahren, deren Zielfunktion[3] konvex ist, also einen konvexen Suchraum aufspannt und somit keine lokalen Optima besitzt, werden als konvex bezeichnet [MPH09]. Konvexe Verfahren liefern die bezüglich ihrer Zielfunktion bestmögliche Einbettung der latenten Punkte, während nichtkonvexe Verfahren diese nicht garantiert finden.

deterministisch – nichtdeterministisch Im Gegensatz zu deterministischen Verfahren, die bei gleicher Eingabe immer das gleiche Ergebnis berechnen, kann das Ergebnis nichtdeterministischer Verfahren von Durchlauf zu Durchlauf variieren.

Eine weitere Kategorisierung ergibt sich, wenn die Verfahren danach unterschieden werden, welche Struktureigenschaften des Datenraums sie gemäß ihres Entwurfs zu erhalten versuchen [MPH07]: Manche Verfahren sind beispielsweise eher distanzbewahrend, während andere auf den Erhalt von Nachbarschaften spezialisiert sind. Zusätzlich kann unterschieden werden, ob diese Eigenschaften global oder nur in lokalen Bereichen der Mannigfaltigkeit erhalten werden. Dimensionsreduktionsverfahren können auch danach eingeteilt werden, welche Methoden, mathematischen Modelle oder Datenstrukturen sie einsetzen, um die Muster einzubetten beziehungsweise, bei parametrischen Verfahren, um die Abbildungsfunktion zu bestimmen. So können die Dimensionsreduktionsverfahren Familien zugeordnet werden, die beispielsweise künstliche neuronale Netze, Regressionsverfahren, Kernfunktionen oder eine Eigenwertzerlegung einsetzen.

[3] Die Zielfunktion ist nicht mit der Abbildungsfunktion zu verwechseln. Parametrische Verfahren bilden mit der Abbildungsfunktion Muster auf latente Punkte ab. Die Abbildungsfunktion wird durch Lösen eines Optimierungsproblems bestimmt, welches bezüglich der Zielfunktion zu optimieren ist. Nichtparametrische Verfahren besitzen keine Abbildungsfunktion und bestimmen daher direkt anhand der Zielfunktion, wie ein Muster optimal eingebettet werden kann.

Die in dieser Arbeit verwendete Systematik zur Ordnung von Dimensionsreduktionsverfahren ist in Abbildung 2.2 zu sehen. Die Verfahren werden nach den von ihnen verwendeten Methoden gruppiert. Sie basiert auf dem Ordnungsschema von [MPH09] und wurde für diese Arbeit angepasst: Weitere nichtkonvexe Familien und Verfahren wurden ergänzt und die Familien allgemeiner eingeteilt. Zur Wahrung der Übersichtlichkeit wird im dargestellten Schema auf die Angabe der weiteren Eigenschaften wie Linearität verzichtet; diese sind in Abschnitt 2.6 zusammengestellt. In dem Schema sind nur die Dimensionsreduktionsverfahren aufgelistet, die in den experimentellen Vergleich dieser Arbeit aufgenommen werden. Die Auswahl der Dimensionsreduktionsverfahren aus einer Vielzahl weiterer wurde so getroffen, dass alle Familien und alle oben genannten Eigenschaften von Dimensionsreduktionsverfahren, wie Linearität oder Konvexität, vertreten sind. Eine Ausnahme bilden die überwachten Dimensionsreduktionsverfahren. Sie werden in dieser Arbeit nicht betrachtet, da sie Labels der Muster benötigen, die jedoch bei realen Anwendungen in der Regel nicht gegeben sind. Die in der Abbildung 2.2 dargestellten Verfahren werden in den folgenden Abschnitten, beginnend bei den konvexen, beschrieben. Die Namen der Verfahren sind englisch belassen, da im deutschen Sprachgebrauch für viele Verfahren keine Übersetzung geläufig ist.

2.3 Konvexe Dimensionsreduktionsverfahren

Bei den in dieser Arbeit vorgestellten konvexen Dimensionsreduktionsverfahren beruht die Optimierung der Zielfunktion auf einer *Eigenwertzerlegung* (auch *Spektralzerlegung*). Als Eigenwertzerlegung wird die Aufgliederung einer quadratischen, symmetrischen Matrix $\mathbf{X}$ in das Produkt

$$\mathbf{X} = \mathbf{V}\mathbf{S}\mathbf{V}^T \tag{2.1}$$

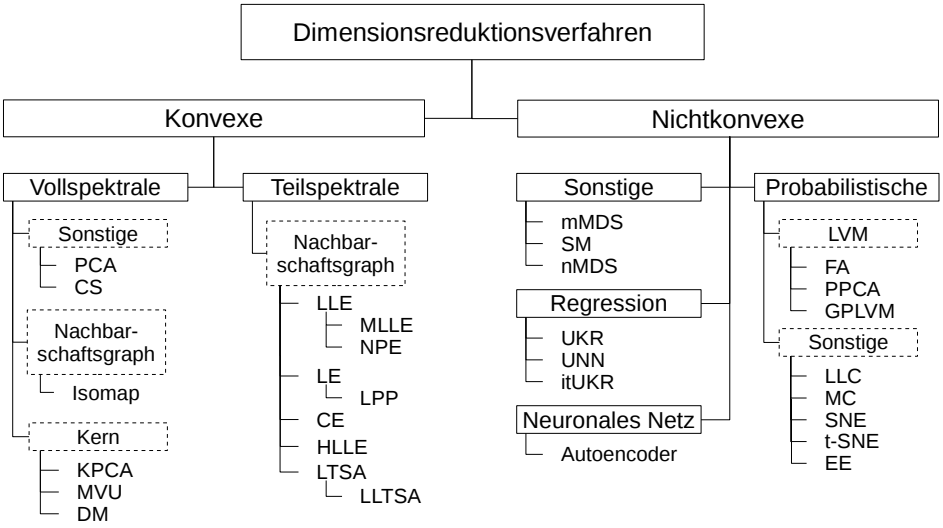


Abb. 2.2: Ordnungsschema für Dimensionsreduktionsverfahren

bezeichnet, wobei $\mathbf{S}$ eine Diagonalmatrix ist, welche die Eigenwerte von $\mathbf{X}$ enthält, die Spalten von $\mathbf{V}$ die Eigenvektoren von $\mathbf{X}$ sind und $\mathbf{V}$ orthogonal ist, das heißt $\mathbf{V}\mathbf{V}^T = \mathbf{V}^T\mathbf{V} = \mathbf{E}$ (mit Einheitsmatrix $\mathbf{E}$) [El ; Mar09].

Verfahren, die auf einer Eigenwertzerlegung basieren, werden gelegentlich auch als *spektrale* Verfahren bezeichnet [XSB06; CPL10]. Sie können danach unterschieden werden, ob sie die Eigenwertzerlegung für eine vollständig oder eine spärlich besetzte Matrix vornehmen.

2.3.1 Vollspektrale Verfahren

Diese Verfahren lösen das Eigenwertproblem für eine Matrix, deren Einträge alle gesetzt sind, und werden daher als *vollspektral* bezeichnet. Es werden zuerst die traditionellen Verfahren PCA und CS beschrieben, anschließend wird auf Isomap eingegangen und zum Schluss werden die kernbasierten Verfahren KPCA, MVU und DM dargestellt.

Principal Components Analysis (PCA)

Beim Entwurf eines Dimensionsreduktionsverfahrens ist zu entscheiden, welche Informationen des Datenraums als wichtig angesehen werden und vom Verfahren nach Möglichkeit in den latenten Raum übertragen werden sollen (siehe Diskussion in Abschnitt 2.1). Das Ziel von PCA [Pea01; Hot33] (zitiert in [Jol02]) ist, möglichst viel Varianz der Muster zu erhalten. PCA interpretiert die Dimensionen des Datenraums als Achsen eines Koordinatensystems und legt ein neues, weniger Achsen besitzendes Koordinatensystem so durch die Daten, dass die Varianz im latenten Raum maximiert wird. Die Achsen sind orthogonal zueinander und folglich unkorreliert. Voraussetzung für PCA ist, dass die Muster zentriert wurden, das heißt dass sie in jeder Dimension den Mittelwert 0 besitzen. Dazu wird von jedem Eintrag einer Spalte in $\mathbf{P}$ der Mittelwert dieser Spalte abgezogen.

PCA nutzt eine Transformationsmatrix, um eine lineare Abbildung der Muster auf latente Punkte vorzunehmen. Im Folgenden wird in Anlehnung an [Mar09; TSK06] beschrieben, wie diese Matrix durch Eigenwertzerlegung bestimmt werden kann. Gesucht ist eine Transformationsmatrix $\mathbf{F}$, welche die Muster in der Matrix $\mathbf{P}$ durch die Matrixmultiplikation

$$\mathbf{L} = \mathbf{P} \cdot \mathbf{F} \tag{2.2}$$

auf ihre latenten Punkte in der Matrix $\mathbf{L}$ abbildet. Die optimale Matrix $\mathbf{F}$ maximiert die Kovarianz der Matrix der latenten Punkte; als Zielfunktion geschrieben:

$$\max \;\; cov(\mathbf{L}), \tag{2.3}$$

wobei $cov(\mathbf{L}) = \mathbf{L}^T \cdot \mathbf{L}$ die Kovarianzmatrix der Matrix $\mathbf{L}$ ist. Diese Matrix ist quadratisch und symmetrisch. Ein Eintrag c_{ij} von $cov(\mathbf{L})$ enthält die Kovarianz der Achsen i und j. Befindet sich c_{ij} in der Diagonale, das heißt ist i

gleich j, entspricht die Kovarianz c_{ij} der Varianz der Achse i bzw. j.

Da die neuen Achsen orthogonal zueinander sein sollen, müssen alle Einträge der Kovarianzmatrix $cov(\mathbf{L})$ mit Ausnahme der Diagonale null sein. Somit enthält die Kovarianzmatrix der latenten Punkte lediglich die Varianzen der Achsen:

$$cov(\mathbf{L}) = \begin{pmatrix} \lambda_1 & 0 & 0 & \cdots & 0 \\ 0 & \lambda_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & \lambda_q \end{pmatrix}. \tag{2.4}$$

Folglich maximiert die Zielfunktion die Varianz des latenten Raums.

Um ein $\mathbf{F}$ zu berechnen, das die Zielfunktion optimiert, muss $cov(\mathbf{L})$ unter Verwendung von Gleichung 2.2 umgeformt werden:

$$\begin{aligned} cov(\mathbf{L}) &= \mathbf{L}^T \cdot \mathbf{L} && (2.5) \\ &= (\mathbf{P} \cdot \mathbf{F})^T \cdot (\mathbf{P} \cdot \mathbf{F}) && (2.6) \\ &= \mathbf{F}^T \cdot \mathbf{P}^T \cdot \mathbf{P} \cdot \mathbf{F} && (2.7) \\ &= \mathbf{F}^T \cdot cov(\mathbf{P}) \cdot \mathbf{F}. && (2.8) \end{aligned}$$

Werden nun $\mathbf{F}$ und $\mathbf{F}^T$ auf die linke Seite gebracht, erhält man

$$\begin{aligned} cov(\mathbf{L}) &= \mathbf{F}^T \cdot cov(\mathbf{P}) \cdot \mathbf{F} \quad | \cdot \mathbf{F} && (2.9) \\ \Leftrightarrow \mathbf{F} \cdot cov(\mathbf{L}) &= cov(\mathbf{P}) \cdot \mathbf{F} \quad | \cdot \mathbf{F}^T && (2.10) \\ \Leftrightarrow \mathbf{F} \cdot cov(\mathbf{L}) \cdot \mathbf{F}^T &= cov(\mathbf{P}). && (2.11) \end{aligned}$$

Gleichung 2.11 entspricht der Eigenwertzerlegung der Matrix $cov(\mathbf{P})$, wenn die Spalten von $\mathbf{F}$ die Eigenvektoren von $cov(\mathbf{P})$ sind und $cov(\mathbf{L})$ eine Diagonalmatrix ist, deren Diagonalelemente die zugehörigen Eigenwerte darstellen. Da $cov(\mathbf{L})$ wie oben beschrieben eine Diagonalmatrix ist (siehe Glei-

chung 2.4), kann die gesuchte Transformationsmatrix $\mathbf{F}$ durch Spektralzerlegung von $cov(\mathbf{P})$ bestimmt werden.

Der Ablauf von PCA lässt sich folgendermaßen zusammenfassen: Zunächst werden die Muster zentriert und die Kovarianzmatrix $cov(\mathbf{P})$ der Muster berechnet. Anschließend wird durch Eigenwertzerlegung der Kovarianzmatrix die Transformationsmatrix $\mathbf{F}$ bestimmt. Die Spalten (Eigenvektoren) von $\mathbf{F}$ werden nach absteigenden Eigenwerten sortiert und nur die Eigenvektoren mit den q größten Eigenwerten bilden die endgültige Transformationsmatrix $\mathbf{F}$. Zum Schluss werden die Muster durch das Produkt $\mathbf{L} = \mathbf{P} \cdot \mathbf{F}$ auf latente Punkte abgebildet.

Classical Scaling (CS)

Classical Scaling [Tor52; Tor58] ist das erste in der Literatur vorgestellte Verfahren des Multidimensional Scaling (MDS). Im Laufe der Zeit wurden viele weitere Varianten des MDS entwickelt, von denen drei in Abschnitt 2.4.1 beschrieben werden. CS gehört mit PCA zu einem der frühesten Dimensionsreduktionsverfahren und ist teilweise auch unter dem Namen *Principal Coordinate Analysis* zu finden [Jol02].

Wie PCA ist CS ein konvexes lineares Dimensionsreduktionsverfahren und liefert, trotz eines anderen Lösungswegs, die gleichen Ergebnisse wie PCA. Statt der Muster $\mathbf{P}$ erhält CS deren euklidische Distanzen d_{ij} in einer $n \times n$ Matrix $\mathbf{D}$ als Eingabe und versucht, diese im latenten Raum möglichst gut zu bewahren. In einer Zielfunktion ausgedrückt bedeutet dies, dass die Abweichungen der Distanzen minimiert werden [MPH09]:

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} (d_{ij}^2 - \delta_{ij}^2), \tag{2.12}$$

wobei $\delta_{ij} = \|\mathbf{l}_i - \mathbf{l}_j\|$ die euklidische Distanz der latenten Punkte $\mathbf{l}_i$ und $\mathbf{l}_j$ ist.

Für die Abbildung der Muster in den latenten Raum wird die sogenannte

Gram-Matrix $\mathbf{K}$ der Muster benötigt. Diese ist durch $\mathbf{K} = \mathbf{P} \cdot \mathbf{P}^T$ definiert [MPH09]. Da CS die Muster $\mathbf{P}$ jedoch nicht als Eingabe erhält, wird ein Eintrag k_{ij} von $\mathbf{K}$ über den Zusammenhang

$$k_{ij} = -\frac{1}{2}\left(d_{ij}^2 - \frac{1}{n}\sum_{l=1}^{n} d_{il}^2 - \frac{1}{n}\sum_{l=1}^{n} d_{lj}^2 + \frac{1}{n^2}\sum_{l=1}^{n}\sum_{m=1}^{n} d_{lm}^2\right) \tag{2.13}$$

aus der Distanz d_{ij} der Muster $\mathbf{p}_i$ und $\mathbf{p}_j$ berechnet [Kra+16]. Dies bewirkt eine „zweifache Zentrierung der [...] quadrierten euklidischen Distanzen“ [MPH09, S. 4]. Da die Distanzen im Daten- und im latenten Raum übereinstimmen sollen, sollten folglich auch die Gram-Matrizen beider Räume möglichst ähnlich sein. Die Zielfunktion kann daher auch anhand der Gram-Matrizen formuliert werden [BG05, S. 262]:

$$\min \ \|\mathbf{L} \cdot \mathbf{L}^T - \mathbf{K}\|^2. \tag{2.14}$$

Aus der Eigenwertzerlegung der Gram-Matrix kann nach [Wil02; Alp10] eine Berechnungsvorschrift für die latenten Punkte abgeleitet werden, welche die Zielfunktion 2.12 optimiert. Dabei sei $\mathbf{S}$ die Diagonalmatrix der Eigenwerte von $\mathbf{K}$ und $\mathbf{V}$ enthalte die zugehörigen Eigenvektoren als Spalten:

$$\begin{aligned}
\mathbf{P} \cdot \mathbf{P}^T = \mathbf{K} &= \mathbf{V} \cdot \mathbf{S} \cdot \mathbf{V}^T && (2.15)\\
&= \mathbf{V} \cdot \mathbf{S}^{\frac{1}{2}} \cdot \mathbf{S}^{\frac{1}{2}} \cdot \mathbf{V}^T && (2.16)\\
&= \mathbf{V} \cdot \mathbf{S}^{\frac{1}{2}} \cdot \left(\mathbf{S}^{\frac{1}{2}}\right)^T \cdot \mathbf{V}^T && (2.17)\\
&= \mathbf{V} \cdot \mathbf{S}^{\frac{1}{2}} \cdot \left(\mathbf{V} \cdot \mathbf{S}^{\frac{1}{2}}\right)^T. && (2.18)
\end{aligned}$$

Hieraus ist ersichtlich, dass $\mathbf{V} \cdot \mathbf{S}^{\frac{1}{2}}$ eine Näherung für $\mathbf{P}$ ist. Bei den Umformungen wurde ausgenutzt, dass die elementweise Quadratwurzel $\mathbf{S}^{\frac{1}{2}}$ der quadratischen Diagonalmatrix $\mathbf{S}$ ebenfall eine quadratische Diagonalmatrix

ist. Folglich ergibt $\mathbf{S}^{\frac{1}{2}} \cdot \mathbf{S}^{\frac{1}{2}}$ wiederum $\mathbf{S}$ und die Transformation $\left(\mathbf{S}^{\frac{1}{2}}\right)^T$ ist identisch zu $\mathbf{S}^{\frac{1}{2}}$.

Werden $\mathbf{V}$ und $\mathbf{S}$ nach absteigenden Eigenwerten sortiert, kann die Matrix der latenten Punkte durch

$$\mathbf{L} = \mathbf{V}_q \cdot \mathbf{S}_q^{\frac{1}{2}} \tag{2.19}$$

berechnet werden, wobei $\mathbf{S}_q^{\frac{1}{2}}$ die q größten Eigenwerte aus $\mathbf{S}$ enthält und $\mathbf{V}_q$ die entsprechenden Eigenvektoren [Wil02]. Laut [MPH09, S. 4] unterstützt die Zielfunktion 2.12 (bzw. 2.14) vor allem die Erhaltung großer Distanzen.

Die Ergebnissse von CS entsprechen denen von PCA; die Begründung kann beispielsweise [Jol02; Wil02; MPH09] entnommen werden. Aus diesem Grund wird CS im experimentellen Vergleich nicht berücksichtigt. CS wurde dennoch beschrieben, da es die Grundlage von auf ihm aufbauenden Varianten (siehe Abschnitt 2.4.1) und von Isomap bildet, das CS verwendet.

Isometric Feature Mapping (Isomap)

Isomap [TDSL00] ist ein nichtlineares Verfahren, das CS zur Dimensionsreduktion verwendet, diesem aber statt euklidischer Distanzen sogenannte *geodätische* Distanzen übergibt, um die Struktur nichtlinearer Mannigfaltigkeiten exakter konstruieren zu können. Dass geodätische Distanzen hierbei euklidischen überlegen sind, ist in der Abbildung 2.3 zu erkennen, welche den nichtlinearen Datensatz Swiss-roll zeigt: Die euklidische Distanz (rot) zweier Muster kann stark von deren geodätischer Distanz (violett) entlang der Struktur der Mannigfaltigkeit abweichen.

Für die Berechnung geodätischer Distanzen wird die Annahme getroffen, dass die euklidische und die geodätische Distanz von im Datenraum nahe benachbarten Mustern nahezu identisch sind. Isomap stellt einen Nachbarschaftsgraph auf, der die Muster als Knoten darstellt und jeden Knoten mit

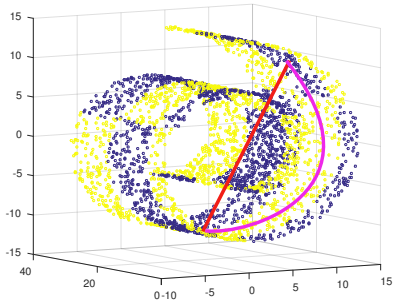


Abb. 2.3: Euklidische (rot) vs. geodätische (violett) Distanz

seinen k nächsten Nachbarn über Kanten verbindet, die entsprechend der euklidischen Distanz gewichtet sind. Die geodätische Distanz nicht benachbarter Muster wird durch Berechnung des kürzesten Pfades zwischen den zugehörigen Knoten im Graph und Summierung der Kantengewichte auf diesem Pfad angenähert. Auf diese Weise werden die paarweisen geodätischen Distanzen aller Muster berechnet und zur Dimensionsreduktion an CS gegeben [Alp10, S. 133].

Zu beachten ist, dass der Nachbarschaftsgraph unter Umständen nicht zusammenhängend ist, sondern aus mehreren nicht miteinander verbundenen Zusammenhangskomponenten besteht. Da geodätische Distanzen nicht für Knoten aus verschiedenen Zusammenhangskomponenten bestimmt werden können, berechnet Isomap in einem solchen Fall nur die Distanzen für Knoten der größten Zusammenhangskomponente und führt auch nur für diese die Dimensionsreduktion durch; die übrigen Muster werden nicht eingebettet [MPH09, S. 18]. Dieser Fall kann ebenfalls bei allen im weiteren Verlauf der Arbeit beschriebenen Verfahren eintreten, die Nachbarschaftsgraphen verwenden.

Kernel PCA (KPCA)

Da lineare Dimensionsreduktionsverfahren im Allgemeinen nichtlineare Strukturen in den Mustern weniger gut erfassen können als nichtlineare Verfahren, wurde KPCA von [SSM98] als nichtlineare Variante von PCA vorgestellt. Durch die Verwendung eines sogenannten *Kerns* soll KPCA die Erhaltung nichtlinearer Strukturen verbessern.

Die Motivation für die Nutzung eines Kerns besteht in der Annahme, dass durch eine Abbildung ϕ der Muster in einen höher-dimensionalen Raum die intrinsische, nichtlineare Struktur der Muster in diesem Raum mit linearen Operationen besser als im ursprünglichen Raum erfasst werden kann [Mar09, S. 125]. Da die Abbildung ϕ und Operationen auf den Punkten des hochdimensionalen Raums aufwändig zu berechnen sind [Mar09, S. 127], wird stattdessen ein Kern verwendet, der im Datenraum auf die Muster angewendet wird und dasselbe Ergebnis erzielt.

Eine Funktion K wird als Kern bezeichnet, wenn sie zwei Punkte eines Raums auf einen Skalar abbildet und dieser dem Skalarprodukt der beiden Punkte entspricht, nachdem diese durch eine *Merkmalsabbildung* ϕ in einen höher-dimensionalen *Merkmalsraum* $\mathcal{H}$ abgebildet wurden [Kra13]. Dies lässt sich folgendermaßen formalisieren

$$K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R} \tag{2.20}$$

$$\phi : \mathbb{R}^d \to \mathcal{H} \tag{2.21}$$

$$K(\mathbf{x}, \mathbf{y}) = \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle, \tag{2.22}$$

wenn $\langle \mathbf{x}, \mathbf{y} \rangle$ das Skalarprodukt zweier Vektoren darstellt. In der Literatur wurden verschiedene Kerne wie beispielsweise der Gauß-Kern vorgestellt; eine Übersicht findet sich in [HSS08]. Für den Anwender hat ein Kern den Vorteil, dass er die meist recht komplexe Funktion ϕ weder kennen noch nutzen muss, sondern stattdessen den Kern direkt auf die ursprünglichen Daten anwenden

kann.

Im Folgenden wird der Ablauf von KPCA in Anlehnung an [MPH09] beschrieben. In KPCA wird zunächst mit Hilfe eines geeigneten Kerns aus den Mustern eine *Kern-Matrix* $\mathbf{K}$ berechnet (auch *Gram-Matrix* [HSS08]). Sie ist quadratisch und symmetrisch, ihre Einträge k_{ij} enthalten das Ergebnis des auf je zwei Muster $\mathbf{p}_i$ und $\mathbf{p}_j$ angewendeten Kerns:

$$k_{ij} = K(\mathbf{p}_i, \mathbf{p}_j). \tag{2.23}$$

Wie bei CS die euklidischen Distanzen, müssen die Einträge der Kern-Matrix $\mathbf{K}$ ebenfalls zweifach zentriert werden. Dazu werden von jedem Eintrag der Matrix der Mittelwert der Zeile und der der Spalte, in welchen sich der Eintrag befindet, abgezogen, der Mittelwert über alle Einträge der Matrix addiert und das Ergebnis mit $-\frac{1}{2}$ multipliziert [MPH09, S. 5]:[4]

$$k_{ij} = -\frac{1}{2}\left(k_{ij} - \frac{1}{n}\sum_{l=1}^{n} k_{il} - \frac{1}{n}\sum_{l=1}^{n} k_{lj} + \frac{1}{n^2}\sum_{l=1}^{n}\sum_{m=1}^{n} k_{lm}\right). \tag{2.24}$$

PCA bedient sich einer Eigenwertzerlegung der Kovarianzmatrix, um eine Transformationsmatrix zur Abbildung der Muster in den latenten Raum zu berechnen. In KPCA werden analog die Eigenvektoren $\mathbf{a}_i$ der Kovarianzmatrix im Merkmalsraum benötigt. Um diese zu berechnen, werden zunächst die Eigenvektoren $\mathbf{v}_i$ und die Eigenwerte λ_i von $\mathbf{K}$ bestimmt. Aus den Eigenvektoren mit den q größten Eigenwerten werden nun über den Zusammenhang

$$\mathbf{a}_i = \frac{1}{\sqrt{\lambda_i}}\mathbf{v}_i \tag{2.25}$$

die Eigenvektoren $\mathbf{a}_i$ der Kovarianzmatrix berechnet. Mit diesen werden ab-

[4] Die Indizes für den Spaltenmittelwert wurden korrigiert.

schließend die latenten Punkte durch die Matrixmultiplikation

$$\mathbf{L} = \mathbf{K} \cdot \mathbf{A} \tag{2.26}$$

berechnet, wenn $\mathbf{A}$ in den Spalten die Eigenvektoren $\mathbf{a}_i$ enthält.

Die Ergebnisse von KPCA hängen vom verwendeten Kern ab. Wenn der lineare Kern

$$K(\mathbf{p}_i, \mathbf{p}_j) = \langle \mathbf{p}_i, \mathbf{p}_j \rangle \tag{2.27}$$

eingesetzt wird, liefern KPCA und CS das gleiche Ergebnis, denn in diesem Fall entspricht die bei CS berechnete Gram-Matrix $\mathbf{K} = \mathbf{P} \cdot \mathbf{P}^T$ (Gleichung 2.13) der Kern-Matrix von KPCA mit $k_{ij} = \langle \mathbf{p}_i, \mathbf{p}_j \rangle$. Andere oft für KPCA verwendete Kerne sind der polynomiale oder der Gauß-Kern [MPH09].

Maximum Variance Unfolding (MVU)

Wie bereits im Abschnitt über KPCA erwähnt, ändert sich das von KPCA erzielte Ergebnis, wenn ein anderer Kern verwendet wird. Dies hat zur Folge, dass in der Praxis getestet werden muss, mit welchem Kern für einen bestimmten Anwendungszweck die besten Ergebnisse erzielt werden. Um diesen zusätzlichen Aufwand zu umgehen, lernt MVU (auch *Semidefinite Embedding*) die Kern-Matrix $\mathbf{K}$, anstatt sie über einen vom Anwender gewählten Kern zu berechnen. Die gelernte Kern-Matrix wird anschließend KPCA zur Dimensionsreduktion übergeben [MPH09].

Um die Kern-Matrix zu lernen, konstruiert MVU, wie Isomap, zunächst einen Nachbarschaftsgraph G, in welchem jedes Muster mit seinen k nächsten Nachbarn verbunden ist und die Gewichte der Kanten der euklidischen Distanz zwischen den jeweiligen Mustern entsprechen [MPH09]. MVU definiert

die Kern-Matrix $\mathbf{K}$ nicht über die Muster, sondern mit dem Skalarprodukt

$$k_{ij} = \langle \mathbf{l}_i, \mathbf{l}_j \rangle \tag{2.28}$$

über die unbekannten latenten Punkte [Kla07] und lernt die Matrix mit dem Ziel, im latenten Raum die Distanzen zu allen k nächsten Nachbarn eines Punktes zu bewahren und alle anderen Distanzen zu maximieren. Auf diese Weise soll die Erhaltung lokaler Strukturen unterstützt werden.

Die Zielfunktion

$$\max \quad \sum_{i=1}^{n} \sum_{j=1}^{n} \|\mathbf{l}_i - \mathbf{l}_j\|^2 \quad = \quad \max \quad \sum_{i=1}^{n} \sum_{j=1}^{n} k_{ij} \tag{2.29}$$

maximiert die Distanzen zwischen allen latenen Punkten, wobei durch

$$\|\mathbf{l}_i - \mathbf{l}_j\|^2 = \|\mathbf{p}_i - \mathbf{p}_j\|^2 \qquad \forall \text{ in } G \text{ benachbarten } (i, j) \tag{2.30}$$

beziehungsweise, mit Hilfe von $\mathbf{K}$ ausgedrückt,

$$k_{ii} - 2k_{ij} + k_{jj} = \|\mathbf{p}_i - \mathbf{p}_j\|^2 \qquad \forall \text{ in } G \text{ benachbarten } (i, j) \tag{2.31}$$

die Veränderung der Distanzen zwischen im Nachbarschaftsgraph benachbarten Mustern verhindert wird [MPH09]. Zusätzlich soll $\mathbf{K}$ zentriert sein. MVU optimiert die Zielfunktion und die Nebenbedingungen mit *semidefiniter Programmierung*, welche zur Optimierung konvexer Zielfunktionen über spezielle, sogenannte *semidefinite* Matrizen verwendet werden kann [Lan+04]. Da die Varianz der Einbettung proportional zur Summe über $\mathbf{K}$ ist, maximiert MVU wie PCA zusätzlich die im latenten Raum bewahrte Varianz [Kla07, S. 46].

Diffusion Maps (DM)

DM basiert auf der Modellierung eines Random-Walk auf den Mustern [CL06] und setzt voraus, dass diese zentriert wurden [Coi+05]. Als Ähnlichkeitsmaß der Muster nutzt DM sogenannte *Verbreitungsdistanzen* (engl. *diffusion distances*), die im latenten Raum erhalten bleiben sollen. Um diese zu berechnen, konstruiert DM einen vollständigen Graph über die Muster, dessen Kantengewichte w_{ij} sich durch Anwendung des Gauß-Kerns ergeben [MPH09]:

$$w_{ij} = e^{-\frac{\|\mathbf{p}_i - \mathbf{p}_j\|^2}{2\sigma^2}}. \tag{2.32}$$

Diese Kantengewichte werden in Transitionswahrscheinlichkeiten $p_{ij}^{(1)}$ umgerechnet, die angeben, mit welcher Wahrscheinlichkeit von einem Muster $\mathbf{p}_i$ nach einem Zeitschritt des Random-Walk ein anderes Muster $\mathbf{p}_j$ erreicht wird. Die Wahrscheinlichkeiten werden in der Matrix $\mathbf{P}^{(1)}$ zusammengefasst. Die Transitionswahrscheinlichkeiten nach t Zeitschritten ergeben sich durch

$$\mathbf{P}^{(t)} = \left(\mathbf{P}^{(1)}\right)^t. \tag{2.33}$$

Aus diesen können die Verbreitungsdistanzen ermittelt werden. Für die Berechnung einer Einbettung, welche die Verbreitungsdistanzen im latenten Raum optimal bewahrt, wird $\mathbf{P}^{(t)}$ durch

$$\mathbf{P}^{(t)}\mathbf{v}_i = \lambda_i \mathbf{v}_i \tag{2.34}$$

in seine Eigenvektoren $\mathbf{v}_i$ und -werte λ_i zerlegt [MPH09]. Mit Hilfe der q größten Eigenwerte und deren zugehörigen Eigenvektoren wird die Matrix der latenten Punkte über

$$\mathbf{L} = (\lambda_2\mathbf{v}_2, \lambda_3\mathbf{v}_3, \cdots, \lambda_{q+1}\mathbf{v}_{q+1}) \tag{2.35}$$

berechnet. Der erste Eigenwert und dessen Eigenvektor werden hierbei nicht berücksichtigt, da sie eine triviale Lösung darstellen [MPH09].

2.3.2 Teilspektrale Verfahren

Die bisher in dieser Arbeit vorgestellten Verfahren führen Eigenwertzerlegungen vollständig besetzter Matrizen aus. Verfahren, welche die Eigenwertzerlegung einer spärlich besetzten Matrix berechnen, werden in dieser Arbeit als *teilspektrale* Verfahren bezeichnet. Die Matrizen besitzen für jede Kombination von Mustern $\mathbf{p}_i$ und $\mathbf{p}_j$ einen Eintrag x_{ij}. Dieser enthält eine Null, wenn die zugehörigen Muster im Datenraum nicht benachbart sind, wobei die Definition der Nachbarschaft dem jeweiligen Dimensionsreduktionsverfahren unterliegt. Da in den meisten Fällen die Nachbarschaftsgröße k um ein Vielfaches kleiner als die Anzahl an Mustern ist, enthalten folglich große Bereiche dieser Matrizen Nullen.

Als teilspektrale Verfahren werden LLE (mit den Varianten MLLE und NPE), LE (mit der Variante LPP), CE, HLLE und LTSA (mit der Variante LLTSA) vorgestellt.

Locally Linear Embedding (LLE)

LLE [RS00] basiert auf den Annahmen, dass nichtlineare Mannigfaltigkeiten in kleinen, lokalen Bereichen eine annähernd lineare Struktur besitzen, durch lokale Anwendung linearer Operationen diese lokalen Strukturen erfasst werden können und daraus die gesamte nichtlineare Mannigfaltigkeit konstruiert werden kann.

Aufgrund dieser Annahmen rekonstruiert LLE zunächst jedes Muster $\mathbf{p}_i$ als Linearkombination seiner k nächsten Nachbarn:

$$\min \quad \sum_{i=1}^{n} \|\mathbf{p}_i - \sum_{j=1}^{n} w_{ij}\mathbf{p}_j\|^2. \tag{2.36}$$

Von der Wahl der Faktoren w_{ij} hängt die Größe der Abweichung ab. Sie werden unter Zuhilfenahme der Methode der kleinsten Quadrate bezüglich dieser Zielfunktion optimiert. Dabei sind die w_{ij} für nicht benachbarte Muster $\mathbf{p}_i$ und $\mathbf{p}_j$ null und die Summe $\sum_j w_{ij}$ ist auf eins beschränkt.

Eine Einbettung wird als gut definiert, wenn deren lokale Strukturen möglichst ähnlich zu denen des Datenraums sind. Wie im Datenraum beschreibt LLE auch im latenten Raum die Struktur der Umgebung eines Punktes $\mathbf{l}_i$ über die Linearkombination seiner k nächsten Nachbarn. So ergibt sich die folgende Zielfunktion zur Optimierung der latenten Punkte:

$$\min \quad \sum_{i=1}^{n} \|\mathbf{l}_i - \sum_{j=1}^{n} w_{ij}\mathbf{l}_j\|^2. \tag{2.37}$$

Die w_{ij} sind die Rekonstruktionsgewichte aus dem Datenraum und stellen die Ähnlichkeit der Strukturen im Daten- und latenten Raum sicher. Denn wenn beispielsweise ein Muster $\mathbf{p}_j$ einen starken Einfluss auf ein Muster $\mathbf{p}_i$ hat, das heißt w_{ij} ist groß, sollte sich dies auch in den zugehörigen latenten Punkten widerspiegeln [Mar09].

Die Matrix $\mathbf{L}$ der bezüglich dieser Zielfunktion optimalen latenten Punkte entspricht laut [RS00] den Eigenvektoren mit den q kleinsten Eigenwerten (ohne Null) der Matrix $\mathbf{B}$, wobei

$$\mathbf{B} = (\mathbf{E} - \mathbf{W})^T(\mathbf{E} - \mathbf{W}) \tag{2.38}$$

gelte, $\mathbf{E}$ die Einheitsmatrix sei und $\mathbf{W}$ die w_{ij} enthalte.

Modified Locally Linear Embedding (MLLE) MLLE [ZW06] wurde als Verbesserung von LLE entwickelt und verwendet ebenfalls Rekonstruktionsgewichte, um mit diesen ein Muster beziehungsweise einen latenten Punkt als Linearkombination seiner k nächsten Nachbarn zu beschreiben. MLLE setzt jedoch nicht nur einen Gewichtsvektor sondern m linear unabhängige

Gewichtsvektoren pro Nachbarschaft ein, sodass sich die Zielfunktion

$$\min \quad \sum_{i=1}^{n} \sum_{z=1}^{m} \|\mathbf{l}_i - \sum_{j=1}^{n} w_{ij}^{(z)} \mathbf{l}_j\|^2 \tag{2.39}$$

ergibt; $w_{ij}^{(z)}$ bezeichnet den Beitrag des latenten Punktes $\mathbf{l}_j$ zum latenten Punkt $\mathbf{l}_i$ entsprechend des Eintrags w_{ij} des z-ten Gewichtsvektors der Nachbarschaft. Die optimale Einbettung wird wie bei LLE durch Eigenwertzerlegung gefunden.

Neighborhood Preserving Embedding (NPE) NPE [He+05] wurde als lineare Variante von LLE entwickelt und hat gleichfalls das Ziel, lokale Strukturen zu erhalten. Laut [Qia+13] wurde davon unabhängig *Neighborhood Preserving Projections* (NPP) [Pan+05] vorgestellt, das äquivalent zu NPE ist. Wie LLE bestimmt NPE zunächst die k nächsten Nachbarn zu jedem Muster und berechnet die optimalen Gewichte w_{ij}, mit denen ein Muster $\mathbf{p}_i$ als Linearkombination seiner Nachbarn rekonstruiert wird. Die Zielfunktion dieser Optimierung ist Gleichung 2.36.

Auch die Zielfunktion zur Berechnung der optimalen latenten Punkte entspricht der von LLE (Gleichung 2.37). In [He+05] wird gezeigt, dass die Eigenvektoren $\mathbf{v}_i$ zu den q kleinsten Eigenwerten, welche die Gleichung

$$\mathbf{PBP}^T\mathbf{v}_i = \lambda_i \mathbf{PP}^T\mathbf{v}_i \tag{2.40}$$

lösen, die Spalten einer Transformationsmatrix $\mathbf{F}$ bilden, mit der über den Zusammenhang

$$\mathbf{L} = \mathbf{P} \cdot \mathbf{F} \tag{2.41}$$

die optimalen latenten Punkte aus den Mustern berechnet werden können. Die Matrix $\mathbf{B}$ entspricht der gleichnamigen, in Gleichung 2.38 von LLE ver-

wendeten Matrix.

Laplacian Eigenmaps (LE)

LE [BN03] strebt die Erhaltung lokaler Strukturen durch Minimierung der Distanzen benachbarter Muster im latenten Raum an. Hierfür wird zunächst der Nachbarschaftsgraph der Muster aufgestellt: Jeder Knoten (Muster) wird mit jedem seiner k nächsten Nachbarn über eine Kante verbunden. Die Kanten werden entsprechend der Distanzen der Muster gewichtet. Zur Berechnung der Gewichte kann beispielsweise der Gauß-Kern verwendet werden [MPH09] (siehe Gleichung 2.32). Auch andere Kerne sind denkbar [LV07, S. 162]. Kleine Distanzen zwischen Mustern $\mathbf{p}_i$ und $\mathbf{p}_j$ resultieren in großen Gewichten w_{ij} [MPH09], sodass bei der Minimierung der Zielfunktion

$$\min \quad \sum_{i=1}^{n} \sum_{j=1}^{n} \|\mathbf{l}_i - \mathbf{l}_j\|^2 w_{ij} \tag{2.42}$$

folglich für die zugehörigen latenten Punkte $\mathbf{l}_i$ und $\mathbf{l}_j$ möglichst kleine Distanzen angestrebt werden.

Um eine Lösung dieser Zielfunktion zu finden, verwendet LE die *Laplace-Matrix* $\mathbf{M}$ des Nachbarschaftsgraphen. Die Laplace-Matrix eines Graphen ergibt sich durch

$$\mathbf{M} = \mathbf{D} - \mathbf{A} \tag{2.43}$$

wenn $\mathbf{D}$ eine Diagonalmatrix ist, deren Einträge d_{ii} den Grad des jeweiligen Knoten i enthalten, und $\mathbf{A}$ die Adjazenzmatrix des Graphen ist. LE erzeugt die Laplace-Matrix in abgewandelter Form: Statt $\mathbf{A}$ wird die aus den Gewichten w_{ij} bestehende Matrix $\mathbf{W}$ verwendet und $\mathbf{D}$ ist eine Diagonalmatrix mit $d_{ii} = \sum_j w_{ij}$. In [BN03] wird gezeigt, dass die Zielfunktion 2.42 unter

Verwendung der Laplace-Matrix äquivalent als

$$\min \quad trace(\mathbf{L}^T\mathbf{M}\mathbf{L}) \tag{2.44}$$

geschrieben werden kann, wobei $trace(\mathbf{X})$ die Summe der Diagonalelemente der Matrix $\mathbf{X}$ bezeichnet.

Die Lösung dieses Optimierungsproblems wird durch Berechnung der Eigenvektoren $\mathbf{v}_i$ und Eigenwerte λ_i, $i \in \{1, \cdots n\}$, für das verallgemeinerte Eigenwertproblem

$$\mathbf{M}\mathbf{v}_i = \lambda_i \mathbf{D}\mathbf{v}_i \tag{2.45}$$

bestimmt [BN03]. Die Matrix $\mathbf{L}$ der latenten Punkte ergibt sich aus den Eigenvektoren zu den q kleinsten Eigenwerten:

$$\mathbf{L} = (\mathbf{v}_2, \cdots, \mathbf{v}_{q+1}) \,. \tag{2.46}$$

Die Eigenvektoren sind dabei nach dem zugehörigen Eigenwert aufsteigend sortiert; der Eigenvektor zum Eigenwert Null wird nicht berücksichtigt.

Locality Preserving Projections (LPP) LPP [HN03] (auch *Laplacianfaces* [Zha+09]) wurde als lineare Variante von LE entwickelt. LPP hat wie LE das Ziel, lokale Nachbarschaften zu bewahren und zeigt auch in der Umsetzung große Ähnlichkeit mit LE. Anders als LE lernt LPP eine Transformationsmatrix $\mathbf{F}$, welche die Muster auf latente Punkte abbildet, und ist somit ein parametrisches Verfahren.

Wie in LE werden ein Nachbarschaftsgraph und dessen Kantengewichte berechnet, auch die Berechnung der Laplace-Matrix $\mathbf{M}$ des Graphen stimmt überein. Die Zielfunktion, die das Auseinanderziehen benachbarter Punkte bestraft, ist ebenalls von LE übernommen (Gleichung 2.44). In [HN03] wird gezeigt, dass die optimale Lösung der Zielfunktion durch die verallgemeinerte

Eigenwertzerlegung

$$\mathbf{PMP}^T\mathbf{v}_i = \lambda_i \mathbf{PDP}^T\mathbf{v}_i \tag{2.47}$$

berechnet werden kann. Werden die Eigenvektoren $\mathbf{v}_i$, welche Gleichung 2.47 lösen, gemäß ihrer Eigenwerte aufsteigend sortiert, ergeben sich die Spalten der Transformationsmatrix $\mathbf{F}$ aus den Eigenvektoren zu den q kleinsten Eigenwerten. Die Matrix der latenten Punkte wird durch Anwendung der Transformationsmatrix auf die Muster berechnet:

$$\mathbf{L} = \mathbf{P} \cdot \mathbf{F}. \tag{2.48}$$

Conformal Eigenmaps (CE)

CE [SS05] überträgt lokale Strukturen des Datenraums durch Erhaltung der Winkel zwischen benachbarten Mustern in den latenten Raum. Dafür berechnet CE zunächst die Eigenvektoren $\mathbf{v}_i$ von LLE oder LE. Für diese werden eine lineare Transformation $\mathbf{F}$ und Skalierungswerte mit Hilfe semidefiniter Programmierung bezüglich eines Fehlermaßes optimiert, das die Erhaltung der Winkel misst. Die latenten Punkte ergeben sich letztendlich durch

$$\mathbf{L} = \mathbf{F} \cdot \mathbf{V}, \tag{2.49}$$

wenn $\mathbf{V}$ spaltenweise die Eigenvektoren $\mathbf{v}_i$ enthält.

Hessian Locally Linear Embedding (HLLE)

HLLE (auch *Hessian Eigenmaps*) [DG03] versucht laut [MPH09], die Krümmung der Struktur, die durch die Muster im hoch-dimensionalen Raum gebildet wird, zu reduzieren. Dazu bestimmt HLLE für jedes Muster $\mathbf{p}_i$ die Nachbarschaft, die aus den bezüglich der euklidischen Distanz k nächsten Nachbarn gebildet wird. Für jede dieser Nachbarschaften berechnet HLLE die

Hesse-Matrix im *Tangentialraum* von $\mathbf{p}_i$. Der Tangentialraum eines Musters $\mathbf{p}_i$ auf der Mannigfaltigkeit ist der Raum, der von den Tangenten aufgespannt wird, die durch diesen Punkt gelegt werden können [BG88, S. 75]. Beispielsweise ist der Tangentialraum eines Punktes auf einer dreidimensionalen Kugel eine Ebene, welche die Kugel in diesem Punkt berührt. Die Hesse-Matrix ist als zweite Ableitung einer Funktion mit gegebenenfalls mehreren Parametern definiert [For13, S. 82]. HLLE nutzt die Hesse-Matrizen der Nachbarschaften, um deren lokale Struktur zu erfassen [DG03]. Aus den einzelnen Hesse-Matrizen wird eine Matrix $\mathcal{H}$ berechnet, die als Maß für die Krümmung der gesamten Struktur der Muster dient. Die Eigenwertzerlegung von $\mathcal{H}$ führt zur optimalen Einbettung: Die Eigenvektoren zu den q kleinsten Eigenwerten (ohne Null) bilden die latenten Punkte. Insgesamt betrachtet ist HLLE sehr ähnlich zu LE, wenn LE statt der Laplace-Matrix die Hesse-Matrix verwenden würde [DG03]. Voraussetzung für die Anwendbarkeit von HLLE ist die Zentrierung der Muster innerhalb der Nachbarschaften. Außerdem muss

$$min(k, n) > q \tag{2.50}$$

erfüllt sein [DG03, S. 5593].

Local Tangent Space Alignment (LTSA)

Auch von LTSA [ZZ04] wird wie in HLLE der lokale Tangentialraum eines Musters und seiner Nachbarschaft als Information über die lokale Struktur der hoch-dimensionalen Daten verwendet. Das Ziel von LTSA ist, die in den lokalen Tangentialräumen berechneten Koordinaten der Muster und ihrer Nachbarn so weit wie möglich zu bewahren, wenn die Koordinaten in das Koordinatensystem des globalen latenten Raumes transformiert werden [ZZ04]. Laut [Zha+09] unterscheiden sich HLLE und LTSA darin, dass HLLE quadratische Transformationen auf dem Tangentialraum vornimmt, LTSA jedoch lineare.

Dazu werden im ersten Schritt von LTSA für jedes Muster dessen bezüg-

lich der euklidischen Distanz k nächste Nachbarn bestimmt. Für jede dieser Nachbarschaften wird eine Korrelationsmatrix erstellt. Auf Grundlage der q größten Eigenvektoren jeder Korrelationsmatrix wird eine Matrix berechnet, deren Eigenvektoren zu den q kleinsten Eigenwerten (ohne Null) die optimale globale Einbettung der Muster ergeben.

Linear Local Tangent Space Alignment (LLTSA) LLTSA [Zha+07] wurde als eine lineare Variante von LTSA entwickelt. Die Zielfunktion und die Vorgehensweise von LLTSA sind sehr ähnlich zu LTSA: Es wird eine Transformationsmatrix berechnet, die mit der zentrierten Matrix der Muster multipliziert wird und auf diese Weise die Muster auf latente Punkte abbildet.

2.4 Nichtkonvexe Dimensionsreduktionsverfahren

Die im nun folgenden Teil dieses Kapitels beschriebenen Dimensionsreduktionsverfahren setzen nichtkonvexe Zielfunktionen für die Optimierung der latenten Punkte ein. Zu Beginn werden Varianten von Multidimensional Scaling vorgestellt, danach probabilistische und auf unüberwachter Regression beruhende Verfahren sowie abschließend ein Verfahren, das ein künstliches neuronales Netz nutzt.

2.4.1 Multidimensional Scaling (MDS)

MDS arbeitet im Gegensatz zu PCA nicht auf den Mustern $\mathbf{P}$, sondern auf einer $n \times n$ Matrix $\mathbf{D}$, welche die paarweisen Ähnlichkeiten d_{ij} aller Muster $\mathbf{p}_i$ und $\mathbf{p}_j$ enthält. Abhängig von der Art dieser Ähnlichkeiten und der konkreten Zielfunktion ergeben sich verschiedene Varianten von MDS. Eine Übersicht ist beispielsweise in [CC00; BG05] zu finden. Das ursprüngliche MDS-Verfahren ist Classical Scaling (CS), welches bereits in Abschnitt 2.3.1

beschrieben wurde. In diesem Abschnitt werden drei nichtlineare, nichtkonvexe Varianten beschrieben: mMDS, SM und nMDS.

Metric MDS (mMDS)

mMDS ist eines der ersten nichtlinearen nichtparametrischen Dimensionsreduktionsverfahren [GH15]. Es erhält wie CS (Abschnitt 2.3.1) Distanzen als Eingabe, die aber nicht euklidisch sein müssen. Im Unterschied zu CS besitzt mMDS eine weniger restriktive Zielfunktion. Während CS die exakte Gleichheit der Distanzen der Muster und latenten Punkte fordert, sieht mMDS vor, dass die Distanzen d_{ij} und δ_{ij} der Muster $\mathbf{p}_i$ und $\mathbf{p}_j$ beziehungsweise der latenten Punkte $\mathbf{l}_i$ und $\mathbf{l}_j$ in einem definierten funktionalen Zusammenhang $d_{ij} \approx f(\delta_{ij})$ stehen [Wil02].

Die Zielfunktion[5] von mMDS wird als Fehlerfunktion über die Abweichung von d_{ij} und $f(\delta_{ij})$ definiert [Wil02]:

$$\min \quad \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} \left(d_{ij} - f(\delta_{ij})\right)^2}{\sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij}^2}. \tag{2.51}$$

Die w_{ij} können zur unterschiedlichen Gewichtung von Abweichungen eingesetzt werden, sodass nichtlineare Strukturen in den Daten berücksichtigt werden können [Kla07]. Der größte Unterschied zur Zielfunktion von CS (Gleichung 2.12) ist die Division. Diese bewirkt gemäß [MPH09], dass große und kleine Distanzen ungefähr den gleichen Anteil zur Zielfunktion beitragen und mMDS somit gegenüber CS mehr Wert auf die Erhaltung kleiner Distanzen legt.

Um die Positionen der latenten Punkte unter Minimierung der Zielfunktion anzuordnen, wird ein *Gradientenabstiegsverfahren* angewendet. Gradienten-

[5] In [QY04, S. 403] ist eine Zielfunktion ohne den Faktor $1/\sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij}^2$ zu finden. Da dort keine Quelle angegeben ist, wurde in dieser Arbeit die Zielfunktion von [Wil02] verwendet. Diese entspricht [CC00, S. 6] (häufig als Referenz für eine Übersicht über MDS-Varianten zitiert), wenn die w_{ij} eins sind.

abstiegsverfahren optimieren die initial gesetzten Parameter einer Zielfunktion, indem sie die Werte der Parameter gegen den Gradienten der Zielfunktion verschieben (bei einem Maximierungsproblem mit dem Gradienten). Gradientenabstiegsverfahren bestimmen nicht garantiert die optimale Lösung, da der Gradientenabstieg in einem lokalen Minimum, das nicht das globale Minimum ist, enden kann. Aus diesem Grund findet mMDS im Gegensatz zu CS nicht zwingend die optimale Lösung. Laut [QY04] sind für mMDS jedoch auch andere, globale Optimierungsstrategien denkbar.

Wird mMDS mit der Zielfunktion 2.51 verwendet, wird auch von *Least-Squares Scaling* gesprochen [Wil02]. Daneben existieren Varianten von mMDS mit anderen Zielfunktionen; eine solche Variante ist Sammon Mapping [Mar09; GH15].

Sammon Mapping (SM)

SM [Sam69] entspricht mMDS bis auf geringfügige Anpassungen der Zielfunktion 2.51. Die Funktion f wird durch die Identitätsfunktion ersetzt [Wil02] und die w_{ij} sind bei SM eins. Zusätzlich wird ein Faktor eingeführt, der laut [MPH09] der Vereinfachung des Gradientenabstiegs dient. Insgesamt ergibt sich nach [MPH09] die Zielfunktion

$$\min \quad \frac{1}{\sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij}} \sum_{i \neq j} \frac{(d_{ij} - \delta_{ij})^2}{d_{ij}}. \tag{2.52}$$

Nonmetric MDS (nMDS)

nMDS wurde von [Kru64a; Kru64b] vorgestellt. Es unterscheidet sich insofern von mMDS, als es nicht auf kontinuierlichen, sondern auf diskreten Ähnlichkeitswerten arbeitet [TSK06, S.713]. Dies können beispielsweise die Ränge der paarweisen Distanzen der Muster sein. mMDS ist daher auch für Anwendungsfälle geeignet, in denen die Muster nicht durch Zahlen repräsentiert

werden oder nur die Ränge der Muster, nicht aber deren absolute Ausprägungen interpretierbar sind [Wil02]. Solche Muster gehören zum nominalen oder ordinalen Skalenniveau. Aus diesem Grund wird nMDS auch als *Ordinal Scaling* [Wil02] bezeichnet.

Für nMDS kann die Zielfunktion von mMDS (Gleichung 2.51) übernommen werden, wobei auch die Distanzen (bzw. Ähnlichkeiten) d_{ij} der Muster durch eine Funktion g transformiert werden müssen [QY04; CC00]:

$$\min \quad \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} \left(g\left(d_{ij}\right) - f\left(\delta_{ij}\right)\right)^2}{\sum_{i=1}^{n} \sum_{j=1}^{n} g\left(d_{ij}\right)^2}. \tag{2.53}$$

Die Funktionen f und g müssen die Ordnung der Ränge der Distanzen bewahren [CC00]. Für die Optimierung der Einbettung bezüglich dieser Zielfunktion können dieselben Techniken wie bei mMDS eingesetzt werden.

2.4.2 Probabilistische Verfahren

Unter probabilistischen Dimensionsreduktionsverfahren werden in dieser Arbeit Verfahren verstanden, die Wahrscheinlichkeitsverteilungen einsetzen. Sie wurden gruppiert in Verfahren, die auf dem *latent variable model* (LVM) oder einem *mixture model* (MM) beruhen, und Verfahren, die Wahrscheinlichkeiten als Maß für die Ähnlichkeit von Mustern und latenten Punkten einsetzen.

LVM ist ein statistischer Ansatz, der in verschiedenen Bereichen wie der Psychologie oder den Sozialwissenschaften zur Schätzung der Wahrscheinlichkeitsverteilung von Daten eingesetzt wird und auch im Rahmen der Dimensionsreduktion angewendet werden kann [Eve84]. LVM interpretiert die beobachteten Merkmale der Muster als Zufallsvariablen (*beobachtete Variablen*), die einer gemeinsamen Wahrscheinlichkeitsverteilung unterliegen. Es wird angenommen, dass die Verteiltung in Wirklichkeit auf einer geringeren Anzahl unbekannter, nicht messbarer Zufallsvariablen, den *latenten Variablen*,[6] be-

[6] Die latenten Variablen sind nicht mit latenten Punkten zu verwechseln; erstere ent-

ruht [Eve84]. Ziel des LVM ist, Werte für die Parameter der Verteilung zu schätzen, sodass die Verteilung die beobachteten Muster möglichst gut erklärt. Dies ist der Fall, wenn die Wahrscheinlichkeit für das Auftreten der beobachteten Muster maximal ist. Wird LVM für die Dimensionsreduktion verwendet, werden nicht nur die Parameter der Wahrscheinlichkeitsverteilung, sondern auch die latenten Punkte geschätzt [GW12].

Für das Schätzen der Parameter (und der latenten Punkte) wird ein *Expectation-Maximization* (EM) Algorithmus [DLR77] verwendet. Dies ist ein iteratives Verfahren, bei welchem im ersten Schritt (expectation) bei geschätzten Parametern für jedes Muster der erwartete latente Punkt berechnet wird und im zweiten Schritt (maximization) die Parameter der Verteilung so angepasst werden, dass die Wahrscheinlichkeit der Muster maximiert wird. Beide Schritte werden solange wiederholt, bis ein gesetzter Schwellwert erreicht ist oder die Änderungen der Werte gering genug sind. Dadurch werden die Parameter der Verteilung so anpasst, dass die Wahrscheinlichkeit für die Muster maximiert wird [Ver04]. Allerdings kann nur die Konvergenz zu einem lokalen Optimum garantiert werden.

Ein MM wird ebenfalls zur Dichteschätzung von Datensätzen eingesetzt [Ver04]. Dabei werden mehrere Dichteschätzer verwendet, deren gewichtete Summe die gesamte Verteilung, das MM, ergibt. Die Parameter des MM werden mittels eines EM-Algorithmus optimiert, sodass die Gesamtwahrscheinlichkeit des Datensatzes maximal ist.

Im Folgenden werden zunächst die LVM-basierten Dimensionsreduktionsverfahren FA, PPCA und GPLVM vorgestellt. Danach werden die auf einem MM beruhenden Verfahren (LLC, MC) und die Verfahren, die Ähnlichkeiten in Form von Wahrscheinlichkeiten darstellen (SNE, t-SNE, EE), beschrieben.

---

sprechen den Achsen bzw. Dimensionen des latenten Raums, letztere sind Punkte im latenten Raum.

Factor Analysis (FA)

FA gehört wie PCA zu den frühen Dimensionsreduktionsverfahren; erste Ideen kamen von [Spe04] aus der Psychologie. Während in PCA die Achsen des latenten Raums Linearkombinationen der Achsen des Datenraums darstellen, geht FA den umgekehrten Weg. Die Achsen des Datenraums (beobachtete Variablen) werden als Linearkombinationen der unbekannten Achsen des latenten Raums (latente Variablen) beschrieben. Seien $\mathbf{o} = (o_1, o_2, \ldots, o_d)$ der Spaltenvektor der d beobachteten Variablen und $\mathbf{f} = (f_1, f_2, \ldots, f_q)$ der Spaltenvektor der q latenten Variablen (auch *Faktoren*). Dann lässt sich der Zusammenhang zwischen beobachteten und latenten Variablen durch

$$\mathbf{o} = \mathbf{F}\mathbf{f} + \boldsymbol{\epsilon} \tag{2.54}$$

beschreiben [Eve84]. Die $d \times q$ Matrix $\mathbf{F}$ enthält sogenannte *Faktorenladungen* (engl. *factor loadings*): Ein Eintrag f_{ij} gibt an, wie viel Einfluss der Faktor f_j auf die beobachtete Variable o_i hat [GH97]. $\boldsymbol{\epsilon} = (\epsilon_1, \epsilon_2, \ldots, \epsilon_d)$ ist ein Spaltenvektor, der für jede beobachtete Variable o_i den Anteil, der durch die latenten Variablen nicht erklärt wird, als Fehlerterm ϵ_i enthält [TSK06].

Der in Gleichung 2.54 gezeigte Zusammenhang lässt sich direkt auf die Muster[7] und die latenten Punkte übertragen, da diese die Ausprägungen der beobachteten beziehungsweise latenten Zufallsvariablen sind:

$$\mathbf{p} = \mathbf{F}\mathbf{l} + \boldsymbol{\epsilon}. \tag{2.55}$$

Ein Muster $\mathbf{p}$ ergibt sich folglich als lineare Transformation des zugehörigen latenten Punktes $\mathbf{l}$.

Die Verteilungen der Zufallsvariablen werden als Normalverteilungen mo-

[7] Die Muster müssen zentriert sein, sodass die Spaltensummen gleich null sind [Mar09, S.235].

delliert [GH97]:

$$\mathbf{f} \sim \mathcal{N}(\mathbf{0}, \mathbf{E}) \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Psi}) \qquad \mathbf{o} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{F}\mathbf{F}^T + \boldsymbol{\Psi}\right). \tag{2.56}$$

Die Faktoren $\mathbf{f}$ sind normalverteilt mit einer Nullmatrix als Mittelwert und der Einheitsmatrix als Kovarianzmatrix. Da angenommen wird, dass die Fehlerwerte $\boldsymbol{\epsilon}$ unabhängig voneinander sind [Eve84], ist die Kovarianzmatrix $\boldsymbol{\Psi}$ ihrer Verteilung als Diagonalmatrix definiert, sodass die Kovarianz der Fehlerwerte unterschiedlicher beobachteter Variablen null ist. Aus diesen Annahmen folgt nach [GH97] die in Gleichung 2.56 angegebene Verteilung der beobachteten Variablen $\mathbf{o}$.

Das Ziel von FA ist, die Parameter $\mathbf{F}$ und $\boldsymbol{\Psi}$ der Verteilungen zu schätzen, um mit ihrer Hilfe auf Grundlage der Muster die erwarteten latenten Punkte zu berechnen. Der Erwartungswert $E(\mathbf{l}_i|\mathbf{p}_i)$ für den latenten Punkt $\mathbf{l}_i$ zum Muster $\mathbf{p}_i$ ist eine lineare Transformation des Musters [GH97]:

$$E(\mathbf{l}_i|\mathbf{p}_i) = \mathbf{F}^T \left(\boldsymbol{\Psi} + \mathbf{F}\mathbf{F}^T\right)^{-1} \mathbf{p}_i. \tag{2.57}$$

Die Parameter und latenten Punkte werden unter Verwendung eines EM-Algorithmus optimiert. Nach der Initialisierung[8] der Parameter $\mathbf{F}$ und $\boldsymbol{\Psi}$ wird im ersten Schritt (E-Schritt) für jedes Muster der erwartete latente Punkt $E(\mathbf{l}_i|\mathbf{p}_i)$ berechnet. Im zweiten Schritt (M-Schritt) werden die Parameter bezüglich der geschätzen latenten Punkte optimiert, sodass die Wahrscheinlichkeit, dass die Muster aus den geschätzten latenten Punkten hervorgegangen sind, maximiert wird. Diese Schritte werden bis zu einem geeigneten Abbruchkriterium wiederholt. Nach der Terminierung des EM-Algorithmus liefert FA latente Punkte, die mindestens ein lokales Optimum darstellen. Die geschätzen Parameter der Verteilungen bilden ein Modell, mit dessen Hilfe Punkte

[8] [Jör67] (zitiert in [Eve84, S. 21]) macht einen Vorschlag für eine nicht zufällige Initialisierung.

des Datenraums eingebettet werden können, die beim Training nicht verwendet wurden.

Probabilistic PCA (PPCA)

PPCA [Row97] hat wie PCA das Ziel, die Varianz des Datenraums zu erhalten, und ist FA im Ansatz und der Umsetzung sehr ähnlich. Die Verteilungen der beobachteten und latenten Variablen sind identisch (siehe Gleichung 2.56), der einzige Unterschied besteht in der Kovarianzmatrix $\mathbf{\Psi}$ der Fehlerterme. Während FA vorschreibt, dass die Fehlerterme der beobachteten Variablen unabhängig voneinander sind und folglich die Kovarianzmatrix diagonal ist, nimmt PPCA an, dass die Kovarianzen für alle Variablen gleich und sehr klein sind. Aus diesem Grund entspricht $\mathbf{\Psi}$ annähernd der Einheitsmatrix $\mathbf{E}$.

Wie bei FA werden die Muster zentriert [Law03, S. 1] und wird ein EM-Algorithmus zur Optimierung der latenten Punkte $\mathbf{L}$ und des Parameters $\mathbf{F}$ genutzt. Die Berechnungsvorschriften für $\mathbf{L}$ und $\mathbf{F}$ sind jedoch aufgrund der abweichenden Definition von $\mathbf{\Psi}$ bei FA und PPCA unterschiedlich. Laut [Row97] konnte gezeigt werden, dass der EM-Algorithmus bei PPCA zum globalen Optimum konvergiert.

Gaussian Process Latent Variable Model (GPLVM)

GPLVM [Law03] ist sehr ähnlich zu PPCA [MPH09] und optimiert ebenfalls die latenten Punkte und die Parameter der Wahrscheinlichkeitsverteilung iterativ. GPLVM unterscheidet sich von PPCA, da es einen Kern verwendet und somit nichtlinear ist [GW12]. Da der Kern über die latenten Punkte und nicht über die Muster berechnet wird, kann GPLVM nicht als probabilistische Variante von KPCA bezeichnet werden [MPH09]. Mit den optimierten Parametern lassen sich laut [GW12, S. 118] weitere Testpunkte einbetten; GPLVM ist folglich ein parametrisches Verfahren.

Locally Linear Coordination (LLC)

LLC [TR02] geht davon aus, dass die Struktur der den Mustern zugrunde liegenden Mannigfaltigkeit in lokalen Bereichen linear angenähert werden kann. Aus diesem Grund führt LLC in einem ersten Schritt ein sogenanntes *mixture of factor analyzers* (MFA) aus. Dieses besteht aus mehreren Instanzen des linearen Dimensionsreduktionsverfahrens FA,[9] die jeweils in einem lokalen Bereich des Datenraums eine lineare Abbildung der Muster in einen lokalen latenten Raum vornehmen.

Ein MFA [GH97] ist ein spezielles MM. Das Konzept der MMs, mehrere Dichteschätzer zur Schätzung der Gesamtdichte heranzuziehen, wird auf MFAs übertragen, indem mehrere FA-Instanzen eingesetzt werden. Im Gegensatz zur ursprünglichen FA, bei der die Muster zentriert und die Mittelwerte der Verteilungen der Zufallsvariablen auf $\mathbf{0}$ gesetzt werden, werden bei einem MFA die Mittelwerte der FA-Instanzen nicht beschränkt, sodass es möglich ist, dass jede Instanz auf einem anderen Bereich des Datenraums arbeitet.

Zur Parameteroptimierung des MFA wird ebenfalls ein EM-Algorithmus verwendet. Im E-Schritt werden für jedes Muster und jede FA-Instanz die Wahrscheinlichkeit, dass die von dieser FA-Instanz repräsentierte Wahrscheinlichkeitsverteilung das Muster erzeugt hat, und der zu erwartende latente Punkt berechnet. Im M-Schritt werden die Parameter der FA-Instanzen (u. a. die Mittelwerte) und der gemeinsamen Wahrscheinlichkeitsverteilungen (u. a. die Faktorenladungen) optimiert.

Die vom MFA erzeugten lokalen latenten Räume werden in einem zweiten Schritt zu einem globalen latenten Raum zusammengeführt. Dabei müssen unter anderem Widersprüche der lokalen Einbettungen aufgelöst werden, die aus überschneidenden Anwendungsräumen der lokalen Dimensionsreduktionsverfahren resultieren können. Im Folgenden wird die Vorgehensweise von LLC detaillierter dargestellt.

[9] Es können stattdessen auch PCA-Instanzen verwendet werden [MPH09].

Die Menge von m FA-Instanzen produziert für jedes Muster $\mathbf{p}_i$ eine Einbettung $\mathbf{z}_{il}$ in den lokalen latenten Raum und einen Wert r_{il}, der ein Maß für die Güte des latenten Punktes ist ($l \in \{1, \cdots m\}$). Aus diesen Werten wird eine $n \times m \cdot d$ Matrix $\mathbf{U}$ mit den Einträgen $\mathbf{u}_{il} = r_{il}\mathbf{z}_{il}$ berechnet [MPH09]. Die Vektoren $\mathbf{u}_{il}$ stellen die mit ihrer Güte gewichteten lokalen latenten Punkte dar [TR02].

Gesucht ist eine lineare Transformation $\mathbf{F}$, welche die gewichteten, lokalen latenten Punkte $\mathbf{u}_{il}$ auf einheitliche globale $\mathbf{l}_i$ abbildet, sodass insgesamt jedes Muster $\mathbf{p}_i$ auf einen latenten Punkt $\mathbf{l}_i$ abgebildet wird. Als Zielfunktion verwendet LLC die von LLE (Gleichung 2.37):

$$\min \quad \sum_{i=1}^{n} \|\mathbf{l}_i - \sum_{j=1}^{n} w_{ij}\mathbf{l}_j\|^2. \tag{2.58}$$

Die benötigten Rekonstruktionsgewichte w_{ij} werden wie bei LLE berechnet.

In [TR02] wird gezeigt, dass die gesuchte Transformationsmatrix $\mathbf{F}$, welche die Zielfunktion 2.58 optimiert, durch die verallgemeinerte Eigenwertzerlegung

$$\mathbf{A}\mathbf{v}_i = \lambda_i \mathbf{C}\mathbf{v}_i \tag{2.59}$$

berechnet werden kann. Hier sind $\mathbf{A} = \mathbf{B}^T\mathbf{U}$ mit $\mathbf{B}$ wie in LLE (siehe Gleichung 2.38) und $\mathbf{C} = \mathbf{U}^T\mathbf{U}$ [MPH09]. Die Eigenvektoren $\mathbf{v}_i$ zu den q kleinsten Eigenwerten (ohne Null) bilden die Transformationsmatrix $\mathbf{F}$, sodass die globalen latenten Punkte mit

$$\mathbf{L} = \mathbf{U}\mathbf{F} \tag{2.60}$$

berechnet werden können.

Obwohl die Abbildung der gewichteten lokalen latenten Punkte $\mathbf{u}_{il}$ auf globale latente Punkte linear ist, ist LLC insgesamt ein nichtlineares Verfahren,

da die $\mathbf{u}_{il}$ nichtlinear aus den $\mathbf{z}_{il}$ und die $\mathbf{z}_{il}$ ihrerseits nichtlinear durch das MFA aus den Mustern berechnet werden. Da das MFA keine konvexe Zielfunktion hat, ist LLC zudem nichtkonvex, obwohl die globale Koordination der lokalen latenten Räume eine konvexe Zielfunktion besitzt [TR02].

Manifold Charting (MC)

Auch MC [Bra03] wandelt die lokalen latenten Räumen, die von einem MFA aus m FA-Instanzen erzeugt wurden, in einen globalen latenten Raum um. MC unterscheidet sich von LLC darin, dass es nicht die Zielfunktion von LLE verwendet. Seien $\mathbf{z}_{il}$ die lokale Einbettung des Musters $\mathbf{p}_i$ durch die j-te FA-Instanz und r_{il} die Güte dieser Einbettung, dann ist

$$\min \quad \sum_{i=1}^{n} \sum_{l=1}^{m} r_{il} \|\mathbf{l}_i - \mathbf{l}_{il}\|^2 \tag{2.61}$$

die Zielfunktion von MC, wobei $\mathbf{l}_{il} = \mathbf{z}_{il}\mathbf{M}$ die Abbildung von $\mathbf{z}_{il}$ in den globalen latenten Raum und $\mathbf{l}_i = \sum_{k=1}^{m} r_{ik}\mathbf{l}_{il}$ die endgültige Einbettung des Musters $\mathbf{p}_i$ in den globalen latenten Raum bezeichnet, bei der alle lokalen Einbettungen berücksichtigt werden [MPH09]. Mit dieser Zielfunktion wird insbesondere bestraft, wenn die lokalen Einbettungen einer FA-Instanz stark von denen einer anderen abweichen. Durch Umformung der Zielfunktion und anschließender Eigenwertzerlegung lässt sich eine Transformation der Muster in globale latente Punkte berechnen.

Stochastic Neighbor Embedding (SNE)

SNE [HR02] ist auf die Erhaltung von Nachbarschaften ausgerichtet. Die Nachbarschaftsstruktur wird in Form bedingter Wahrscheinlichkeiten beschrieben, die im Daten- und latenten Raum übereinstimmen sollen. Die bedingte Wahrscheinlichkeit $p_{j|i}$ gibt an, mit welcher Wahrscheinlichkeit ein Muster $\mathbf{p}_i$ das Muster $\mathbf{p}_j$ zum Nachbarn „wählt“ [MH08, S. 2581]. Für die Berechnung

dieser Wahrscheinlichkeit wird bei jedem Muster eine Normalverteilung zentriert. Entsprechend der Dichte dieser Verteilung wird $p_{j|i}$ größer, je dichter $\mathbf{p}_i$ und $\mathbf{p}_j$ beieinander liegen. Analog werden die Wahrscheinlichkeiten $q_{j|i}$ für zwei latente Punkte $\mathbf{l}_i$ und $\mathbf{l}_j$ definiert.

Das Ziel von SNE ist, die Abweichungen der $p_{j|i}$ und $q_{j|i}$ zu minimieren. Dazu wird mit

$$C = \sum_{i=1}^{n} \sum_{j=1}^{n} p_{j|i} \log \frac{p_{j|i}}{q_{j|i}} = \sum_{i=1}^{n} \mathrm{KL}(P_i \| Q_i) \tag{2.62}$$

eine Kostenfunktion aufgestellt, welche die Kullback-Leibler-Divergenz[10] (KL) zwischen den aus $p_{j|i}$ und $q_{j|i}$ hervorgehenden Verteilungen P_i und Q_i definiert [HR02]. Die optimalen latenten Punkte werden berechnet, indem die Kostenfunktion mit Hilfe eines Gradientenabstiegsverfahrens minimiert wird. Dabei werden die initialen latenten Punkte zufällig erzeugt, sodass SNE nichtdeterministisch ist.

t-Distributed SNE (t-SNE)

t-SNE [MH08] wurde als Verbesserung von SNE entwickelt. Nachteile von SNE bestehen darin, dass die Kullback-Leibler Divergenz nicht symmetrisch ist und dadurch bestimmte Fehlplatzierungen latenter Punkte weniger als andere bestraft werden. Beispielsweise wird stärker bestraft, wenn die latenten Punkte benachbarter Muster weit voneinander entfernt platziert werden, als wenn die latenten Punkte entfernter Muster benachbart angeordnet werden. t-SNE behebt diese Schwäche, indem die bedingten Wahrscheinlichkeiten $p_{j|i}$ mit

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n} \tag{2.63}$$

[10] Diese gibt an, wie unterschiedlich zwei Wahrscheinlichkeitsverteilungen sind.

in symmetrische umgewandelt werden [Maa14].

Ein weiterer Nachteil von SNE, den auch andere Dimensionsreduktionsverfahren besitzen, ist die zu nahe Anordnung latenter Punkte, sodass sich Ballungsgebiete latenter Punkte bilden können [MH08]. Um dies zu umgehen, nutzt t-SNE statt der Normalverteilung eine Student-t-Verteilung, um für die latenten Punkte die Wahrscheinlichkeiten q_{ij} zu berechnen.

Gemäß dieser Anpassungen wird die zu minimierende Kostenfunktion über die symmetrischen Wahrscheinlichkeiten p_{ij} und q_{ij} beziehungweise deren gemeinsame Verteilungen P und Q formuliert [Maa14]:

$$C = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}} = \mathrm{KL}(P \| Q). \tag{2.64}$$

Die latenten Punkte werden wie bei SNE mittels eines Gradientenabstiegsverfahrens optimiert; neue Testpunkte können jedoch wie bei SNE nicht eingebettet werden, da t-SNE nichtparametrisch ist [MH08, S. 2600].

Elastic Embedding (EE)

EE [CP10] wurde als Verbesserung von SNE entwickelt, um unter anderem die bei SNE auftretende zu nahe Platzierung latenter Punkte zu beheben. Dazu greift EE Elemente von SNE und LE auf. Die Zielfunktion enthält einen Term, welcher der Zielfunktion von LE entspricht und die Erhaltung lokaler Distanzen unterstützt, und einen additiven zweiten Term für die Erhaltung globaler Distanzen, der Ähnlichkeiten zur Zielfunktion von SNE aufweist. Laut [CP10] besitzt der Gradient dieser Zielfunktion weniger lokale Optima als der der Zielfunktion von SNE, sodass EE insgesamt einfacher optimiert werden kann. Zusätzlich wurde ein spezieller Algorithmus zur Optimierung entwickelt, sodass EE nahezu deterministisch ist.

2.4.3 Unüberwachte Regressionsverfahren

In diesem Abschnitt werden zunächst die Prinzipien überwachter und unüberwachter Regression gegenübergestellt. Anschließend werden drei Dimensionsreduktionsverfahren vorgestellt, die unüberwachte Regression nutzen.

Regression bezeichnet die Vorhersage von Merkmalen für Muster. Regressionsverfahren schätzen die Ausprägung des vorherzusagenden Merkmals eines Musters $\mathbf{p} \in \mathbb{R}^d$ über einen funktionalen Zusammenhang (die *Regressionsfunktion*)

$$f : \mathbb{R}^d \to \mathbb{R}^q. \tag{2.65}$$

Ist $q > 1$, ist das vorherzusagende Merkmal mehrdimensional (*multivariat*).

Überwachte Regressionsverfahren werden zur Vorhersage von beobachtbaren Merkmalen von Mustern eingesetzt [Mei+05]. Sie arbeiten auf einem Trainingsdatensatz, der Muster enthält, für die das vorherzusagende Merkmal beobachtet wurde. Auf dieser Datengrundlage berechnet ein solches Verfahren für ein gegebenes Muster, bei dem keine Information über dieses Merkmal vorliegt, mit Hilfe der Regressionsfunktion die Ausprägung des Merkmals. Ein möglicher Ansatz ist, dem Merkmal den Wert der durchschnittlichen Merkmalsausprägung der k ähnlichsten Trainingsmuster zuzuweisen; dies ist die KNN-Regression.

Im Gegensatz dazu sagen *unüberwachte* Regressionsverfahren unbeobachtbare Merkmale für Muster vorher [Mei+05]. Beobachtbare Merkmale können in der Realität erfasst werden. Da dies in machen Anwendungen beispielsweise aus Zeit- oder Kostengründen nicht geschieht, werden überwachte Regressionsverfahren eingesetzt, um die fehlenden Merkmalsausprägungen vorherzusagen. Nicht beobachtbare Merkmale hingegen sind künstlich und können in der Realität nicht beobachtet werden. Auch die Muster der Trainingsdatensätze verfügen nicht über diese Merkmale.

Dimensionsreduktionsverfahren nutzen unüberwachte Regression, um eine kompaktere Darstellung der Muster zu erzeugen, wenn $q < d$. Sie vertauschen in der Regressionsfunktion Ein- und Ausgabe, sodass mit

$$f_{\mathbf{L}}(\mathbf{l}) : \mathbb{R}^q \to \mathbb{R}^d, \quad f_{\mathbf{L}}(\mathbf{l}) = \hat{\mathbf{p}} \tag{2.66}$$

ein beobachtetes Muster aus einem unbekannten, nicht beobachtbaren latenten Punkt $\mathbf{l}$ rekonstruiert wird. Die Dimensionsreduktionsverfahren versuchen, die optimalen latenten Punkte so zu bestimmen, dass die rekonstruierten Muster $\hat{\mathbf{p}}$ möglichst wenig von den ursprünglichen Mustern abweichen. Die Güte der Rekonstruktion wird mit dem *Datenraumrekonstruktionsfehler* (DSRE)

$$\mathrm{DSRE}(\mathbf{L}) = \|\mathbf{P} - f_{\mathbf{L}}(\mathbf{L})\|_F^2 \tag{2.67}$$

gemessen [Kra13]; $f_{\mathbf{L}}(\mathbf{L})$ ist eine Matrix, welche die Rekonstruktionen aller Muster enthält. Die Dimensionsreduktionsverfahren passen den DSRE teilweise geringfügig an.

In den nächsten Abschnitten werden drei auf unüberwachter Regression beruhende Dimensionsreduktionsverfahren vorgestellt: UKR, UNN und itUKR.

Unsupervised Kernel Regression (UKR)

UKR [Mei+05] wurde aus der überwachten Kernel Regression abgeleitet. Die Kernel Regression ist ein statistischer, nichtlinearer Regressionsansatz, der die Ein- und Ausgabedaten als Ausprägungen von Zufallsvariablen interpretiert, wobei die Ausprägungen der vorherzusagenden, abhängigen Variable als Ausgabedaten und die Ausprägungen der unabhängigen Variable als Eingabedaten bezeichnet werden [Kla07]. Die Zufallsvariablen können ein- oder mehrdimensional sein. Die erwartete Ausgabe zu einer Eingabe ist der bedingte Erwartungswert, der mit Hilfe eines Kerndichteschätzers der gemeinsamen

Verteilung der Variablen berechnet wird [Mei+05]. Ein oft verwendeter Kerndichteschätzer ist der *Nadaraya-Watson-Schätzer*.

UKR wurde als unüberwachte Variante der Kernel Regression mit dem Nadaraya-Watson-Schätzer entworfen. Die Eingabe von UKR ist ein unbekannter, latenter Punkt und die Ausgabe das zugehörige Muster, das mittels des Regressionsmodells

$$f_{\mathbf{L}}(\mathbf{l}) = \sum_{i=1}^{n} \frac{K(\mathbf{l} - \mathbf{l}_i)}{\sum_{j=1}^{n} K(\mathbf{l} - \mathbf{l}_j)} \mathbf{p}_i \tag{2.68}$$

aus den latenten Punkten rekonstruiert wird [Mei+05]; K ist ein wählbarer Kern. Das Ziel von UKR ist, wie bei den übrigen Regressionsverfahren, latente Punkte zu bestimmen, für welche der DSRE minimal ist:

$$\min \quad \frac{1}{n} \sum_{i=1}^{n} \|\mathbf{p}_i - f_{\mathbf{L}}(\mathbf{l}_i)\|^2. \tag{2.69}$$

Die latenten Punkte werden über ein Gradientenabstiegsverfahren bezüglich der Zielfunktion optimiert, nachdem sie durch ein anderes Dimensionsreduktionsverfahren wie PCA oder MDS initialisiert wurden.

Unsupervised Nearest Neighbors (UNN)

UNN ist wie UKR ein unüberwachtes Regressionsverfahren, das von [Kra13] zusammen mit verschiedenen Varianten vorgeschlagen wurde. Im Folgenden wird die Variante der Gauß'schen Einbettungen vorgestellt.

Wie UKR bettet UNN die Muster so ein, dass der DSRE minimal ist, unterscheidet sich aber in zwei Aspekten von UKR. Zum einen verwendet UNN kein Gradientenabstiegsverfahren, sondern bettet die Muster iterativ ein, sodass der DSRE in jeder Iteration minimal verschlechtert wird. Zum anderen setzt UNN das Regressionsmodell *k-nächste-Nachbarn* (KNN) statt des Nadaraya-Watson-Schätzers ein.

Der Ablauf von UNN ist wie folgt: Für das erste einzubettende Muster wird der latente Punkt beliebig gewählt (z. B. ein Nullvektor) und in die in jeder Iteration wachsende Matrix $\overline{\mathbf{L}}$ der latenten Punkte eingefügt. In den folgenden Iterationen wird für das jeweils einzubettende Muster $\mathbf{p}_i$ eine Menge von κ möglichen latenten Punkten berechnet und der beste als endgültiger latenter Punkt ausgewählt.

Dafür wird zunächst das Muster $\mathbf{p}^*$ bestimmt, das unter den m bereits eingebetteten Mustern $\mathbf{p}_i$ bezüglich der quadrierten euklidischen Distanz am ähnlichsten ist:

$$\mathbf{p}^* = \arg \min_{\mathbf{p}=\mathbf{p}_1,\ldots,\mathbf{p}_m} \|\mathbf{p}_i - \mathbf{p}\|^2. \tag{2.70}$$

Die κ möglichen latenten Punkte für $\mathbf{p}_i$ werden berechnet, indem auf den latenten Punkt $\mathbf{l}^*$ des Musters $\mathbf{p}^*$ ein normalverteilter Zufallswert $\mathbf{z}_l$ addiert wird:

$$\mathbf{l}_l^* = \mathbf{l}^* + \mathbf{z}_l\,, \quad \mathbf{z}_l \sim \|\mathbf{p}_i - \mathbf{p}^*\|^2 \cdot \mathcal{N}(0,1) \qquad \forall\, l \in \{1,\ldots,\kappa\} \tag{2.71}$$

Für jeden dieser κ Kandidaten wird das zugehörige Muster mit Hilfe des KNN-Regressionsmodells

$$f_{\overline{\mathbf{L}}}(\mathbf{l}_l^*) = \frac{1}{k} \sum_{j \in \mathcal{N}_k\left(\mathbf{l}_l^*, \overline{\mathbf{L}}\right)} \mathbf{p}_j \tag{2.72}$$

rekonstruiert. $\mathcal{N}_k\left(\mathbf{l}_l^*, \overline{\mathbf{L}}\right)$ ist die Indexmenge der k nächsten Nachbarn des Kandidaten $\mathbf{l}_l^*$. Die Nachbarn sind latente Punkte bereits eingebetter Muster und $\mathbf{p}_j$ ist das zugehörige Muster eines solchen Nachbarn.

Zusätzlich wird für jeden Kandidaten $\mathbf{l}_l^*$ die Abweichung des aus ihm rekonstruierten Musters zum tatsächlichen Muster $\mathbf{p}_i$ berechnet:

$$\mathbf{e}_{\overline{\mathbf{L}}}(\mathbf{l}_l^*) = \|f_{\overline{\mathbf{L}}}(\mathbf{l}_l^*) - \mathbf{p}_i\|^2. \tag{2.73}$$

Dies ist eine Abwandlung des DSRE zur Anwendung auf einzelne latente Punkte. Der Kandidat, für den die Abweichung am geringsten ist, wird als latenter Punkt für $\mathbf{p}_i$ in $\overline{\mathbf{L}}$ übernommen. Diese Schritte werden wiederholt, bis alle Muster eingebettet sind.

Das Ziel von UNN ist die Erhaltung von Nachbarschaften und Distanzen. Nachbarschaften werden laut [Kra15] durch die Optimierung des DSRE bewahrt. Die Distanzen zwischen Mustern werden in den latenten Raum übertragen, da die Zufallsgröße $\mathbf{z}_l$ (Gleichung 2.71) einer Normalverteilung unterliegt, die mit dem Abstand zwischen dem einzubettenden Muster und seinem nächsten Nachbarn skaliert ist.

Iterative UKR (itUKR)

In dieser Arbeit wird das ursprüngliche Verfahren von itUKR vorgestellt, wie es in [Lüc15] beschrieben ist. Erweiterungen sind beispielsweise in [LK14; LK15] zu finden.

itUKR geht aus UKR und UNN hervor, indem es von UNN die iterative Einbettung der Muster übernimmt und das KNN-Regressionsmodell durch den Nadaraya-Watson-Schätzer von UKR ersetzt. Ein Unterschied zu UNN besteht in einer abweichenden Bestimmung des nächsten Musters $\mathbf{p}^*$ zu dem einzubettenden Muster $\mathbf{p}_i$. Statt wie bei UNN (Gleichung 2.70) den quadrierten euklidischen Abstand zu allen bereits eingebetteten Mustern zu berechnen, bestimmt itUKR den Abstand zu den Rekonstruktionen der Muster:

$$\mathbf{p}^* = \arg \min_{\mathbf{l}=\mathbf{l}_1,\ldots,\mathbf{l}_m} \|\mathbf{p}_i - f_{\overline{\mathbf{L}}}(\mathbf{l})\|^2. \tag{2.74}$$

Die übrigen Schritte werden analog zu denen von UNN ausgeführt.

2.4.4 Künstliche neuronale Netze

Die Nutzung eines künstlichen neuronalen Netzes (ANN)[11] zur Dimensionsreduktion führt zu einer weiteren Familie von Dimensionsreduktionsverfahren. Ein ANN besteht aus einer Eingabe- und einer Ausgabeschicht von künstlichen Neuronen, dazwischen können sich weitere Neuronenschichten befinden. Die Neuronen sind über gewichtete Verbindungen mit den Neuronen der vorherigen und der nachfolgenden Schicht verbunden. Die Eingaben, die ein Neuron erhält, sind entweder die an das ANN angelegten Eingabedaten oder, falls sich das Neuron nicht in der ersten Schicht des Netzes befindet, die Ausgabe der Neuronen der Vorgängerschicht. Jedes Neuron verknüpft die Eingaben, die entsprechend der Verbindungsgewichte gewichtet werden, linear oder nichtlinear und gibt sie an jedes Neuron der nächsten Schicht weiter. Insgesamt kann ein ANN auf einer Eingabe komplexe, nichtlineare Operationen ausführen. Für eine weiterführende Einführung in ANNs siehe beispielsweise [Bis95; Alp10]. Als ein Vertreter dieser Familie wird der Autoencoder im nächsten Abschnitt erläutert.

Dimensionsreduktionsverfahren, die *self-organizing maps* (SOMs) einsetzen (z. B. *Generative Topographic Mapping* (GTM) [Sve98; BSW98]), können ebenfalls der Familie von Verfahren zugeordnet werden, die auf ANNs basieren, da auch SOMs das Konzept künstlicher Neuronen verwenden. Diese Verfahren führen nicht nur eine Dimensionsreduktion durch, sondern clustern die Muster zusätzlich, sodass mehreren Mustern dasselbe Neuron (d. h. derselbe latente Punkt) zugeordnet wird. Da in der vorliegenden Arbeit für vergleichbare Evaluationsergebnisse Verfahren gewünscht sind, die für jedes Muster einen eigenen latenten Punkt berechnen, werden auf SOMs basierende Verfahren in dieser Arbeit nicht berücksichtigt.

[11] ANN ist die Abkürzung der englischen Bezeichnung *artificial neural network*. Sie wird in der vorliegenden Arbeit statt der deutschen verwendet, um eine Überschneidung mit der Bezeichnung des KNN-Regressionsmodells zu verhindern.

Autoencoder

Ein Autoencoder [DC92; HS06] ist ein ANN mit mehreren Schichten von Neuronen. Die Anzahl der Neuronen in der Ein- und der Ausgabeschicht entspricht der Dimensionalität d des Datenraums [MPH09]. Zwischen diesen beiden Schichten liegt eine ungerade Anzahl von Schichten, deren Mitte eine Schicht mit q Neuronen bildet. Wird ein Muster an den Autoencoder angelegt und durch die Schichten propagiert, entsprechen die Werte der Neuronen dieser mittleren Schicht dem latenten Punkt des Musters. Der Autoencoder ist symmetrisch aufgebaut, das heißt die Schichten vor und nach der mittleren Schicht sind symmetrisch und die Verbindungsgewichte invers, sodass aus dem latenten Punkt, wenn er von der mittleren Schicht aus weiter durch das Netz propagiert wird, die Werte der Neuronen der Ausgabeschicht wieder dem eingegebenen Muster entsprechen.

Diese Eigenschaft wird beim Training der Gewichte des Autoencoders ausgenutzt: Die Gewichte werden mit zufälligen Werten initialisiert und dahingehend optimiert, dass die Abweichung des rekonstruierten Musters vom eingegebenen Muster möglichst gering ist [HS06]. Für die Optimierung wird laut [MPH09; HS06] ein Trainingsverfahren verwendet, das in lokalen Minima enden kann.

2.5 Erweiterungen von Dimensionsreduktionsverfahren

Die in den vorangegangenen Abschnitten erläuterten Dimensionsreduktionsverfahren wurden zumeist in ihrer ursprünglichen Form vorgestellt. Neben den Varianten, die spezifisch für einzelne Verfahren entwickelt wurden, existieren generelle Erweiterungsmöglichkeiten von Dimensionsreduktionsverfahren, die auf die meisten Verfahren angewendet werden können.

Viele Dimensionsreduktionsverfahren können unter Verwendung eines Kerns

neu formuliert werden (z. B. Isomap, LE, LLE [Ham+03]) oder zu nichtlinearen Verfahren umgewandelt werden (z. B. NPE [He+05], LPP [HN03]). Sogenannte *landmark*-Varianten sind für sehr große Datensätze konzipiert. Sie passen das ursprüngliche Dimensionsreduktionsverfahren so an, dass die Dimensionsreduktion nur für eine Teilmenge der Muster durchgeführt werden muss (siehe beispielsweise [ST03] für MDS und Isomap oder [MH08] für t-SNE). *Out-of-sample*-Erweiterungen sind eine dritte Anpassungsmöglichkeit. Sie werden verwendet, um auch bei nichtparametrischen Dimensionsreduktionsverfahren nachträglich Muster einbetten zu können, nachdem für die übrigen Muster des Datensatzes bereits latente Punkte erzeugt wurden [MPH09]. Bei parametrischen Verfahren ist eine solche Erweiterung nicht notwendig, da die Verfahren die Transformation vom Daten- in den latenten Raum explizit lernen und folglich auch neue Muster einbetten können.

Ein weiterer Typ von Dimensionsreduktionsverfahren sind sogenannte *submanifold learner*, die nicht nur eine Mannigfaltigkeit sondern mehrere mit gegebenenfalls unterschiedlicher Dimensionalität oder Struktur lernen [Kra13]. Zusätzlich lassen sich aus der Kombination mehrerer Dimensionsreduktionsverfahren neue, sogenannte *hybride* Dimensionsreduktionsverfahren konstruieren. Beispiele hierfür sind die von [Kra+16] vorgeschlagenen Hybride MDS-PCA, MDS-LLE und LLE-PCA oder die Kombination von MDS mit SOMs von [WC05].

In den experimentellen Vergleich dieser Arbeit werden Verfahren mit den hier genannten Erweiterungen nicht aufgenommen, um dessen Umfang sinnvoll zu begrenzen. Die out-of-sample-Erweiterung bildet eine Ausnahme. Sie wird gegebenenfalls bei Verfahren eingesetzt, die Nachbarschaftsgraphen verwenden, da diese nur die Muster einbetten, die sich in der größten Zusammenhangskomponente des Graphen befinden. Falls der Graph nicht zusammenhängend ist, werden in dieser Arbeit die übrigen Muster mit Hilfe der out-of-sample-Erweiterung nachträglich eingebettet.

2.6 Eigenschaften der Dimensionsreduktionsverfahren

Für einen besseren Überblick über die vorgestellten Dimensionsreduktionsverfahren werden deren Eigenschaften in diesem Abschnitt tabellarisch zusammengefasst: Tabelle 2.1 enthält die konvexen Verfahren, Tabelle 2.2 die nichtkonvexen. Verfahren unterschiedlicher Familien sind durch gestrichelte Linien getrennt. Die Tabellen enthalten die in Abschnitt 2.2 genannten Eigenschaften und zusätzlich Angaben über die Berechnungskomplexität und darüber, welche Informationen die Dimensionsreduktionsverfahren als wichtig ansehen und bei der Abbildung der Muster in den latenten Raum zu erhalten versuchen. In diesem Abschnitt werden außerdem Begründungen für die angegebenen Eigenschaften von Verfahren gegeben, falls diese nicht bereits in den beschreibenden Texten erwähnt wurden oder sich nicht aus diesen erschließen lassen. Darüber hinaus werden die Quellen für die Komplexitätswerte angegeben.

Verfahren	Linear?	Parametrisch?	Komplexität	Bewahrte Information
PCA	ja	ja	$O(d^3)$	Varianz
CS	ja	nein	$O(n^3)$	(große) euklidische Distanzen
Isomap	nein	nein	$O(n^3)$	(große) geodätische Distanzen
KPCA	nein	nein	$O(n^3)$	Varianz
MVU	nein	nein	$O((nk)^3)$	Varianz, lokale Distanzen
DM	nein	nein	$O(n^3)$	(kleine) Verbreitungsdistanzen
LLE	nein	nein	$O(pn^2)$	Rekonstruktionsgewichte
MLLE	nein	nein	$O(qn^2 + k^3n)$	Rekonstruktionsgewichte
NPE	ja	ja	$O(pn^2 + ndq)$	Rekonstruktionsgewichte
LE	nein	nein	$O(pn^2)$	lokale euklidische Distanzen
LPP	ja	ja	$O(pn^2)$	lokale euklidische Distanzen
CE	nein	nein	$O(q^9)$	lokale Winkel
HLLE	nein	nein	$O(kn^2)$	lokale Krümmung
LTSA	nein	nein	$O(pn^2)$	lokale Struktur
LLTSA	ja	ja	$O(pn^2 + ndq)$	lokale Struktur

Tab. 2.1: Eigenschaften der konvexen Dimensionsreduktionsverfahren

Verfahren	Linear?	Parametrisch?	Komplexität	Bewahrte Information
mMDS	nein	nein	$O(n^2)$	(alle) euklidischen Distanzen
SM	nein	nein	$O(in^2)$	(alle) euklidischen Distanzen
nMDS	nein	nein	$O(n^2)$	Ähnlichkeiten
FA	ja	ja	$\geq$ PCA	Wahrscheinlichkeitsverteilung
PPCA	ja	ja	$O(qnd)$	Varianz
GPLVM	nein	ja	$O(n^3)$	Wahrscheinlichkeitsverteilung
LLC	nein	ja	$O(imq^3)$	Wahrscheinlichkeitsverteilung
MC	nein	ja	$O(imq^3)$	Wahrscheinlichkeitsverteilung
SNE	nein	nein	$O(n^2)$	Nachbarschaften
t-SNE	nein	nein	$O(n^2)$	Nachbarschaften
EE	nein	nein	$O(iqn^2)$	Nachbarschaften
UKR	nein	nein	$O(idn^2)$	DSRE
UNN	nein	ja	$O(n^2)$	Nachbarschaften, Distanzen (DSRE)
itUKR	nein	ja	$O(n^2)$	DSRE
Autoenc.	nein	ja	$O(inw)$	DSRE

Tab. 2.2: Eigenschaften der nichtkonvexen Dimensionsreduktionsverfahren

Die in den Tabellen aufgelisteten Eigenschaften der Verfahren werden im Folgenden begründet.

parametrisch – nichtparametrisch Den in [Zha+07, S. 1549] durchgeführten Evaluationsstudien ist zu entnehmen, dass PCA, NPE, LPP und LLTSA auch auf Muster anwendbar sind, die nicht zum Trainingsdatensatz gehören; es sind folglich parametrische Verfahren. UNN und itUKR können ebenfalls als parametrisch bezeichnet werden, da sie eine explizite Abbildungsfunktion (die Regressionsfunktion) vom latenten Raum in den Datenraum definieren. Die Angaben für PCA, CS, Isomap, KPCA, MVU, DM, LLE, LE, HLLE, LTSA, SM, LLC, MC und Autoencoder sind [MPH09, S. 15] entnommen.

konvex – nichtkonvex Eine Besonderheit unter den nichtkonvexen Verfahren ist PPCA: Es besitzt zwar eine nichtkonvexe Zielfunktion, aber laut [Row97] konnte gezeigt werden, dass PPCA zum globalen Optimum konvergiert.

deterministisch – nichtdeterministisch Alle in dieser Arbeit beschriebenen nichtkonvexen Verfahren sind nichtdeterministisch. Die meisten, da sie die Initialwerte des EM-Algorithmus (FA, PPCA, GPLVM, LLC, MC), des Gradientenabstiegsverfahrens (mMDS, SM, nMDS, SNE, t-SNE, EE, UKR) oder des Trainingsprozesses (Autoencoder) zufällig setzen. UNN und itUKR sind nichtdeterministisch, da sie zur Bestimmung möglicher latenter Punkte Zufallswerte erzeugen. Die konvexen Vefahren hingegen sind deterministisch. Aus diesem Grund ist für diese Eigenschaft keine Spalte in den Tabellen enthalten.

Komplexität Die Komplexitätsangaben für PCA, CS, Isomap, KPCA, MVU, DM, LLE, LE, HLLE, LTSA, SM, LLC, MC und den Autoencoder wurden

[MPH09] entnommen, auch die Variablenbezeichner wurden teilweise übernommen: i bezeichnet die Anzahl an Iterationen, p gibt den Anteil gesetzter Einträge in einer spärlich besetzten Matrix an und w ist die Anzahl an Gewichten im Autoencoder. Die Komplexität der folgenden Verfahren konnte ebenfalls direkt aus der Literatur entnommen werden: PPCA [Row97, S. 630], GPLVM [Law05, S. 1795], SNE [GH15, S. 56], t-SNE [MH08, S. 2600], UKR [Mei+05, S. 1390], UNN [Kra15, S. 1655]. Bei den übrigen Verfahren sind keine eindeutigen Angaben zu finden; für diese werden die in den Tabellen angegebenen Komplexitäten kurz begründet.

LPP ist als lineare Variante von LE diesem sehr ähnlich. Die Eigenwertzerlegung einer spärlich besetzten Matrix ist die aufwändigste Operation und wird bei [MPH09] mit $O(pn^2)$ angegeben. Daher wird LPP in diese Klasse eingeordnet.

Der Ablauf von NPE ist nahezu identisch zu dem von LLE, allerdings kommt die Matrixmultiplikation hinzu, um die latenten Punkte mit Hilfe der Transformationsmatrix zu berechnen. Es werden eine $n \times d$ und eine $d \times q$ Matrix multipliziert, die Komplexität beträgt also $O(ndq)$. Folglich erhöht sich die Komplexität von NPE gegenüber der von LLE auf $O(pn^2 + ndq)$.

Bei CE kommt gegenüber LLE beziehungsweise LE der Aufwand für das Lösen des semidefiniten Programms hinzu. Die Autoren von CE geben keine konkrete Komplexitätsklasse von CE an, sondern weisen darauf hin, dass mit dem semidefiniten Programm ein kleines zusätzliches Problem gelöst werde [SS05]. [GM12, S. 21] zeigt, dass semidefinite Programme in polynomieller Zeit gelöst werden können und [PG10] gibt die Komplexität von CE mit $O(q^9)$ an. Da $O(q^9)$ größer als die Komplexität von LLE beziehungsweise LE ist, wird $O(q^9)$ in die Tabelle übernommen.

Wie bei NPE kommt bei LLTSA gegenüber seinem nichtlinearen Äquivalent LTSA der Aufwand für die Matrixmultiplikation hinzu, welche die Muster in latente Punkte transformiert. Daher wird die Komplexität von LLTSA analog zu NPE auf $O(pn^2 + ndq)$ gesetzt.

Für mMDS und nMDS wird eine quadratische Komplexität angenommen, da auch alle anderen Dimensionsreduktionsverfahren, die einen Gradientenabstieg einsetzen, einen quadratischen Aufwand besitzen.

Für FA wurde keine Komplexitätsklasse gefunden. Die einzige konkretere Aussage ist [HDR97, S. 70] zu entnehmen: Demzufolge ist der Berechnungsaufwand von FA höher als der von PCA, weil die Ausführung des EM-Algorithmus aufwändiger als eine Eigenwertzerlegung ist.

Für EE ist in [CP10, S. 171] der Aufwand für eine Iteration des Algorithmus mit $O(qn^2)$ angegeben. In Abhängigkeit von der Iterationszahl ergibt sich folglich $O(iqn^2)$.

Zu itUKR ist ebenfalls keine Komplexität angegeben. Da itUKR jedoch bis auf das verwendete Regressionsmodell UNN gleicht, kann dieselbe Komplexitätsklasse angenommen werden.

Bewahrte Information Die gleichnamige Spalte in den Tabellen gibt an, welche Informationen des Datenraums die Dimensionsreduktionsverfahren gemäß ihres Entwurfs bei der Einbettung der Muster in den latenten Raum präferiert übernehmen. Diese Angaben basieren auf theoretischen Annahmen der Autoren der Verfahren. Die Angabe „(große) Distanzen“ sagt aus, dass große Distanzen gegenüber kleineren bevorzugt erhalten werden, „(kleine) Distanzen“ bedeutet das Gegenteil und „(alle) Distanzen“ meint, dass große und kleine Distanzen mit gleicher Präferenz bewahrt werden. Als „lokale Distanzen“ werden die Distanzen zu den benachbarten Mustern bezeichnet.

3 Experimenteller Vergleich der Dimensionsreduktionsverfahren

In diesem Kapitel wird zunächst das Evaluationskonzept vorgestellt (Abschnitt 3.1), anhand dessen die im vorigen Kapitel beschriebenen Dimensionsreduktionsverfahren miteinander verglichen werden. Anschließend werden die Ergebnisse hinsichtlich der konvexen (Abschnitt 3.2) und der nichtkonvexen Verfahren (Abschnitt 3.3) jeweils untereinander verglichen und abschließend gemeinsam ausgewertet (Abschnitt 3.4).

3.1 Evaluationskonzept

Das Evaluationskonzept legt die Metriken (Abschnitt 3.1.1) und Datensätze (Abschnitt 3.1.2) fest, anhand derer die Dimensionsreduktionsverfahren verglichen werden. Die verwendeten Implementierungen der Verfahren werden in Abschnitt 3.1.3 beschrieben. Abschnitt 3.1.4 geht darauf ein, wie die Parameter der Verfahren für die Experimente eingestellt wurden.

3.1.1 Metriken

Die Metriken wurden unter Zuhilfenahme des *Goal-Question-Metric*-Paradigmas [BGR94] ausgewählt. Dies ist ein top-down Ansatz, bei welchem ausgehend von einem oder mehreren Untersuchungszielen Fragen formuliert werden, mit Hilfe derer die Zielerreichung gemessen werden kann. Diesen Fragen

werden jeweils eine oder mehrere Metriken zugeordnet, welche die Fragen beantworten.

Das Ziel des experimentellen Vergleichs dieser Arbeit ist, die Dimensionsreduktionsverfahren hinsichtlich verschiedener, durch Metriken quantifizierbarer Eigenschaften miteinander zu vergleichen (Ziel 1). Die zugehörigen Fragen und die sie beantwortenden Metriken werden im Folgenden erläutert, wobei die Fragen Eigenschaften der Dimensionsreduktionsverfahren widerspiegeln und die Metriken deren Ausprägungsgrad messen. Wie bereits in Abschnitt 2.1 beschrieben, existiert für Eigenschaften guter Mannigfaltigkeiten kein einheitliches Kriterium und infolgedessen auch nicht für Dimensionsreduktionsverfahren. Stattdessen versuchen die Verfahren, die Mannigfaltigkeit eines Datensatzes durch Erhaltung topologischer Information anzunähern. Folglich können die Verfahren danach bewertet und verglichen werden, wie gut sie bestimmte Strukturen des Datenraums erhalten. Dem Goal-Question-Metrics-Ansatz folgend ergibt sich daraus die Frage „*Ist das Verfahren topologieerhaltend?*“. Da die Struktur einer Mannigfaltigkeit durch verschiedene Aspekte charakterisiert werden kann, wird diese Frage in drei Teilfragen mit zugehörigen Metriken zerlegt. Als Metriken, die spezielle Struktureigenschaften messen, werden ENX („*Werden Nachbarschaften bewahrt?*“) und EKS („*Werden Distanzen bewahrt?*“) gewählt. Die dritte Teilfrage („*Werden Strukturen bewahrt?*“) dient der Untersuchung der Strukturerhaltung im Allgemeinen; zugeordnete Metriken sind E1NN, DSRE und CRR.

Neben der Topologieerhaltung wird in dieser Arbeit untersucht, wie gut sich die Verfahren als Vorverarbeitung der Daten für eine Klassifikation oder Regression eignen („*Ist das Verfahren informationserhaltend?*“); die zugehörigen Metriken sind EKNN und CRR. Außerdem werden die visualisierten Einbettungen der Verfahren miteinander verglichen, um die Frage „*Eignet sich das Verfahren zur Visualisierung?*“ zu beantworten. Diese Frage wird nicht durch Metriken, sondern lediglich durch Betrachtung der Einbettungen beantwortet. Da diese subjektiv ist und von der Erfahrung des Betrachters abhängt

[LMS12], wird dieser Frage weniger Relevanz als den übrigen zugemessen.

Ein weiteres Ziel des experimentellen Vergleichs ist, die Eigenschaften der Verfahren bei unterschiedlichen Datensätzen zu vergleichen (Ziel 2). Dieses Ziel wird dadurch erreicht, dass die Verfahren auf mehreren Datensätzen ausgeführt und die obigen Metriken auf den jeweiligen Ergebnissen berechnet werden; somit werden keine weiteren Metriken benötigt. Allerdings können CRR und E1NN nur bei diskreten und EKNN nur bei kontinuierlichen Labels berechnet werden. ENX, DSRE und EKS sind bei allen Datensätzen anwendbar.

In den folgenden Abschnitten werden die Berechnungsvorschriften und Bedeutungen der Metriken erläutert. Zunächst werden die Metriken betrachtet, welche die Topologieerhaltung messen (ENX, EKS, DSRE, E1NN), und anschließend die Metriken, welche die Informationserhaltung messen (CRR, EKNN).

ENX

Der ENX misst die Nachbarschaftserhaltung, indem für jedes der n Muster berechnet wird, wie viele der zu den k nächsten Mustern gehörigen latenten Punkte ebenfalls benachbart sind. Diese Werte werden summiert und mit nk normalisiert [Lue+11]:

$$\mathrm{QNX}\,(k) = \frac{1}{nk} \sum_{i=1}^{k} \sum_{j=1}^{k} q_{ij}. \tag{3.1}$$

q_{ij} ist ein Eintrag der sogenannten *Co-Ranking Matrix* Q. Diese wurde von [LV09] vorgestellt und kann als Basis verschiedener Metriken angesehen werden. Die Co-Ranking Matrix wird auf Grundlage der distanzbasierten Ränge der Muster und latenten Punkte erstellt. Für jedes Muster $\mathbf{p}_s$ wird bezüglich

jedes anderen Musters $\mathbf{p}_r$ sein Rang mit

$$\rho_{rs} = |\{t \,|\delta_{rt} < \delta_{rs} \vee (\delta_{rt} = \delta_{rs} \wedge 1 \leq t < s \leq n)\}| \tag{3.2}$$

berechnet [Lue+11]. δ_{ij} bezeichnet die euklidische Distanz zwischen zwei Mustern $\mathbf{p}_i$ und $\mathbf{p}_j$ und $|\cdot|$ die Kardinalität einer Menge. Analog werden die Ränge der latenten Punkte mit

$$\varrho_{rs} = |\{t \,|d_{rt} < d_{rs} \vee (d_{rt} = d_{rs} \wedge 1 \leq t < s \leq n)\}| \tag{3.3}$$

berechnet, wenn d_{ij} der Abstand zweier latenter Punkte $\mathbf{l}_i$ und $\mathbf{l}_j$ ist. Aus diesen Rängen ergeben sich die Einträge der Co-Ranking Matrix mit

$$q_{ij} = |\{(r,s) \,|\rho_{rs} = i \wedge \varrho_{rs} = j\}|. \tag{3.4}$$

Für diese Arbeit wird dem Vorschlag von [Lue+11] folgend der QNX mit

$$\mathrm{ENX}\,(k) = 1 - \mathrm{QNX}\,(k) \tag{3.5}$$

zu einem Fehlermaß umgewandelt, d. h. kleinere Werte sind besser als größere, damit die Werte der verschiedenen Metriken besser vergleichbar sind. Insgesamt besitzt der ENX den Wertebereich $\mathrm{ENX} \in [0, 1]$.

In [LV09] werden weitere distanzbasierte Ränge nutzende Metriken mit Hilfe der Co-Ranking Matrix formuliert: *Trustworthiness* und *Continuity* [VK01; VK06], die in der Vergleichsstudie von [MPH09] verwendet werden, *Local Continuity Meta-Criterion* und *Mean Relative Rank Errors*. Auch die Metriken *Precision* und *Recall*, die aus dem Information Retrieval stammen und in abgewandelter Form in der Zielfunktion eines von [Ven+10] vorgeschlagenen Dimensionsreduktionsverfahrens zur Visualisierung von Datensätzen eingesetzt werden, können nach [Mok+13] mit der Co-Ranking Matrix definiert werden. Diese werden in der vorliegenden Arbeit nicht eingesetzt, da der

ENX für alle Nachbarschaftsrößen sinnvolle Information liefert, während die Aussagekraft der übrigen Metriken teilweise unklar ist oder bestimmte Fehler in den Einbettungen übersehen werden [LV09].

EKS

Der EKS (*Kruskal-Shepard-Scaling* [Kra+16]) ist ein Maß für die Bewahrung der Distanzen und entspricht der Zielfunktion von CS beziehungsweise MDS [HTF09, S. 570] (siehe Abschnitte 2.3.1 und 2.4.1 der vorliegenden Arbeit). In dieser Arbeit wird die von [Kra+16] vorgestellte, normalisierte Variante

$$\text{EKS} = \|\mathbf{D_P} - \mathbf{D_L}\|_F^2 \tag{3.6}$$

verwendet, wobei $\mathbf{D_P}$ die normalisierte Distanzmatrix der Muster und $\mathbf{D_L}$ die der latenten Punkte ist. Ein Eintrag d_{ij} einer Distanzmatrix $\mathbf{D}$ ist der euklidische Abstand der Punkte i und j. $\mathbf{D}$ wird normalisiert, indem jeder Eintrag durch den Wert des größten Matrixelementes dividiert wird. Dies ist notwendig, um die EKS-Werte verschiedener Dimensionsreduktionsverfahren miteinander vergleichen zu können [Kra+16]. Der EKS nimmt Werte aus dem Intervall $[0, \infty($ an; Null ist der beste Wert.

DSRE

Der DSRE bewertet, wie gut die ursprünglichen Muster mit Hilfe des KNN-Regressionsmodells $f_\mathbf{L}$ aus den latenten Punkten rekonstruiert werden können (siehe Gleichungen 2.67 und 2.72):

$$\text{DSRE}(\mathbf{L}, k) = \|\mathbf{P} - f_\mathbf{L}(\mathbf{L})\|_F^2. \tag{3.7}$$

Bei UNN und itUKR stellt der DSRE ein Informationserhaltungsmaß dar, da das KNN-Modell der von UNN und itUKR verwendeten Rekonstruktionsfunktion vom latenten Raum in den Datenraum entspricht. So ist die Ab-

weichung zwischen Mustern und Rekonstruktionen ein direktes Maß, wie viel Information durch die Dimensionsreduktion verloren wurde.

Soll der DSRE auch bei den anderen Verfahren als Informationserhaltungsmaß dienen, darf in der Formel nicht das KNN-Regressionsmodell als Rekonstruktionsfunktion verwendet werden. Stattdessen müssen die Verfahren eine Rekonstruktion vom latenten Raum in den Datenraum entweder explizit angeben (so z. B. beim Autoencoder) oder die Abbildung vom Daten- in den latenten Raum muss invertierbar sein, was insbesondere bei den nichtparametrischen Verfahren nicht gegeben ist [GH15]. Aus diesem Grund dient der DSRE (mit dem KNN-Regressionsmodell) in dieser Arbeit nicht als Informationserhaltungsmaß. Stattdessen ist ein guter, d. h. kleiner DSRE-Wert lediglich ein Anzeichen dafür, dass die latenten Punkte ähnlicher Muster nahe beieinander liegen. Denn eine gute Rekonstruktion eines Musters $\mathbf{p}$ ergibt sich dann, wenn es aus ihm sehr ähnlichen Mustern rekonstruiert wird. Dazu müssen diese Muster zu den latenten Punkten gehören, welche die Nachbarn des zu $\mathbf{p}$ gehörigen latenten Punktes sind. Der DSRE wird daher in dieser Arbeit als Qualitätsmaß der Struktur der Mannigfaltigkeiten interpretiert.

E1NN

Der E1NN bewertet ebenfalls die Struktur der Mannigfaltigkeiten. Er wurde von [San08] vorgeschlagen und misst die Überlappung von latenten Punkten verschiedener Klassen, indem die Anzahl l^- an latenten Punkten, deren nächster Nachbar ein Label einer anderen Klasse trägt, ins Verhältnis zur Gesamtanzahl n der latenten Punkte gesetzt wird:

$$E_{\mathbf{L}} = \frac{l^-}{n}. \tag{3.8}$$

Um den Wertebereich des E1NN an den des EKNN anzupassen, wurde der $E_{\mathbf{L}}$ für diese Arbeit durch den $E_{\mathbf{P}}$ dividiert, der analog zum $E_{\mathbf{L}}$ die Überlappung

der Muster verschiedener Klassen misst:

$$\text{E1NN} = \frac{E_{\mathbf{L}}}{E_{\mathbf{P}}} = \frac{l^-}{p^-}. \tag{3.9}$$

Ist der E1NN größer als eins, ist die Struktur der Mannigfaltigkeit schlechter als die Struktur im Datenraum, ist er gleich eins, sind beide gleich gut, und ist der E1NN kleiner als eins, ist die Struktur der Mannigfaltigkeit besser.

CRR

Der CRR (*Classification Rate Reduction*) wurde von [NS03] vorgestellt und für diese Arbeit ähnlich wie der E1NN zu

$$\text{CRR}\,(k) = \frac{\text{Anzahl falsch klassifizierter latenter Punkte}}{\text{Anzahl falsch klassifizierter Muster}} \tag{3.10}$$

angepasst. Es wird die KNN-Klassifikation verwendet, die einem Muster (bzw. latenten Punkt) das unter den k nächsten Mustern (bzw. latenten Punkten) am häufigsten vorkommende Label zuweist.

Der CRR ist im Gegensatz zum DSRE für alle Dimensionsreduktionsverfahren ein Informationserhaltungsmaß. Er stuft ein Dimensionsreduktionsverfahren als informationserhaltend ein, wenn die Klassifikation der latenten Punkte nahezu dasselbe Ergebnis wie eine Klassifikation der Muster hat, der CRR also möglichst klein ist. Der CRR ist folglich ein Maß dafür, wie gut sich ein Verfahren als Vorverarbeitung der Muster für eine Klassifikation eignet.

Laut [GH15] können auf Klassifikation beruhende Metriken, und somit auch der CRR, als Strukturerhaltungsmaße verwendet werden, wenn künstlichen Datensätzen Labels so hinzugefügt werden, dass diese an die Struktur der Mannigfaltigkeit angepasst sind. Dies ist bei den in dieser Arbeit verwendeten künstlichen Datensätzen gegeben.

EKNN

Der EKNN [Kra+16] ist wie der CRR ein Informationserhaltungsmaß, kann jedoch nur bei Mustern mit kontinuierlichen Labels angewendet werden:

$$\mathrm{EKNN}\,(k) = \frac{\text{Regressionsfehler im latenten Raum}}{\text{Regressionsfehler im Datenraum}}. \tag{3.11}$$

Es wird die KNN-Regression verwendet, die jedem Muster (bzw. latenten Punkt) den Mittelwert der Labels seiner k nächsten Nachbarn zuweist. Der EKNN gibt Auskunft darüber, wie gut die latenten Punkte für eine anschließende Weiterverarbeitung mit Regressionverfahren geeignet sind. Je kleiner der Wert, desto besser ist die Einbettung.

Weitere Metriken

Neben den bisher betrachteten Metriken existieren in der Literatur weitere Vorschläge für Metriken zur Bewertung von Dimensionsreduktionsverfahren. Ansätze, die aufgrund fehlender zugänglicher Implementierungen nicht verwendet werden, sind *Normalization Independent Embedding Quality Assessment* (NIEQA) [ZRZ12], das ein lokales und ein globales Qualitätsmaß zur Bewertung von Nachbarschaften beziehungsweise globalen Strukturen bereitstellt, und die von [RL15] vorgestellte Technik zur Visualisierung der Eigenschaften von Einbettungen mit Hilfe einer sogenannten *Data Description Landscape*.

Außerdem kann aus der Zielfunktion jedes Dimensionsreduktionsverfahrens eine Metrik konstruiert werden [Lue+11]. So entstandene Metriken können die Evaluationsergebnisse verzerren, da die nutzenden Dimensionsreduktionsverfahren automatisch bevorzugt werden. In der vorliegenden Arbeit sind solche Metriken der EKS, der in der Zielfunktion der MDS-Varianten enthalten ist, und der DSRE, der von UKR, UNN und itUKR optimiert wird. Dies wird bei der Auswertung berücksichtigt.

3.1.2 Datensätze

Um die Eigenschaften der Dimensionsreduktionsverfahren bei möglichst verschiedenartigen Datensätzen untersuchen zu können, wurden reale Datensätze aus unterschiedlichen Anwendungsbereichen und künstliche Datensätze ausgewählt. Bei der Auswahl der realen Datensätze wurde auf eine möglichst heterogene Datenraumdimensionalität geachtet, die von gering bis sehr hoch reicht. Es kommen nur gelabelte Datensätze zur Anwendung, da einige Metriken das Vorhandensein von Labels voraussetzen. Die Eigenschaften der gewählten, nach aufsteigender Dimensionalität sortierten realen Datensätze sind in Tabelle 3.1 dargestellt. Die intrinsische Dimensionalität dieser Datensätze wurde geschätzt, indem auf den jeweiligen Datensatz, bzw. eine Teilmenge von 3000 Mustern, der Maximum-Likelihood-Schätzer aus der Matlab-Toolbox von [Maa] angewendet wurde. Da dieser Schätzer für MNIST keine gültige Dimensionalität berechnet, wurde die intrinsische Dimensionalität von MNIST mit dem ebenfalls aus dieser Toolbox stammenden GMST-Schätzer bestimmt. Die Datensätze sind folgenden Quellen entnommen: Iris, Boston und Digits [Ped+11], RCT (Relative Location of CT[1] Slices) und MNIST [Lic], HIVA [Hiv], CNS (Central Nervous System) [Hen+] und ORL [Orl].

[1] Computertomographie (CT): medizinische Aufnahmetechnik zur Darstellung von Hartgewebe (z. B. Knochen, Tumoren)

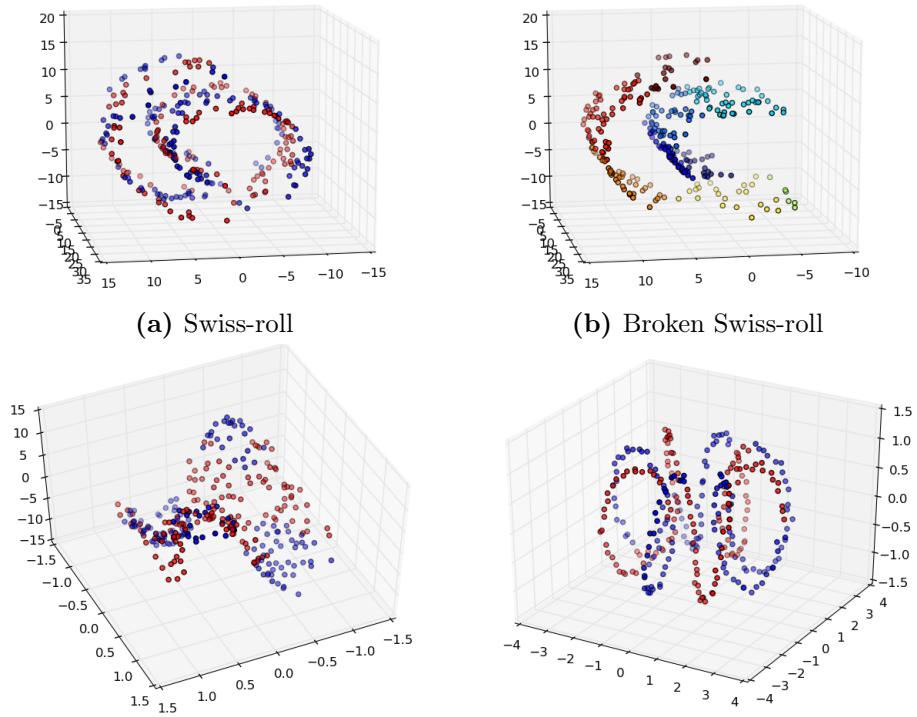
	Dim.	Intrins. Dim.	#Muster	Art d. Labels (#Klassen)	Beschreibung
Iris	4	3	150	diskret (3)	Blüteneigenschaften von Schwertlilien zur Vorhersage der Art
Boston	13	2	506	kontin. (-)	Wohnungsdaten zur Vorhersage von Hauspreisen in Boston
Digits	64	17	1797	diskret (10)	Bilder handgeschriebener Zahlen zur Vorhersage der abgebildeten Zahl
RCT	386	28	53500	kontin. (-)	CT-Bilder verschiedener Personen zur Vorhersage der Aufnahmeposition
MNIST	784	2	60000	diskret (10)	Bilder handgeschriebener Zahlen zur Vorhersage der abgebildeten Zahl
HIVA	1617	15	3845	diskret (2)	Eigenschaften von Wirkstoffen zur Vorhersage ihrer Wirksamkeit gegen HIV
CNS	7130	30	60	diskret (2)	Gendaten von Tumorpatienten zur Vorhersage, ob Patienten überleben werden
ORL	10304	7	400	diskret (40)	Bilder von Gesichtern verschiedener Personen zur Gesichtserkennung

Tab. 3.1: Eigenschaften der realen Datensätze

Die fünf in dieser Arbeit verwendeten künstlichen Datensätze sind dieselben wie in [MPH09]: Swiss-roll, Broken Swiss-roll und Twin-peaks sind dreidimensionale Datensätze (siehe Abbildung 3.1) mit einer zweidimensionalen Mannigfaltigkeit, Helix ist ebenfalls dreidimensional, besitzt aber eine eindimensionale intrinsische Struktur. HD ist ein zehndimensionaler Datensatz mit fünfdimensionaler Mannigfaltigkeit. Die Datensätze wurden mit der Matlab-Toolbox [Maa] generiert. Dabei wurden den Mustern diskrete Labels so hinzugefügt, dass die aufgefaltete Mannigfaltigkeit ein Schachbrettmuster besitzt. Den Mustern von Broken Swiss-roll wurden stattdessen Labels aus zehn verschiedenen Klassen zugewiesen.

Mit Hilfe der künstlichen Datensätze wird überprüft, ob die Dimensionsreduktionsverfahren Datensätze mit Mannigfaltigkeiten spezieller Eigenschaften verarbeiten können. Mit Broken Swiss-roll wird das Verhalten der Verfahren bei nicht zusammenhängenden Mannigfaltigkeiten und mit HD das Verhalten bei einer Mannigfaltigkeit untersucht, die gegenüber den anderen künstlichen Datensätzen eine hohe intrinsische Dimensionalität besitzt. Helix und Twin-peaks wurden ausgewählt, da ihre Daten auf einer Mannigfaltigkeit liegen, die nicht isometrisch zum euklidischen Raum ist [MPH09]. Im Gegensatz zum Datensatz Swiss-roll, dessen zweidimensionale Mannigfaltigkeit sich durch Auseinanderwickeln der Rolle ergibt, entsteht die Mannigfaltigkeiten von Twin-peaks durch „Herunterdrücken“ der Peaks, sodass die geodätischen Distanzen nicht mehr dieselben sind: Die Entfernung eines Punktes, der auf einem Peak der Mannigfaltigkeit liegt, entlang der Mannigfaltigkeit zu einem Punkt, der in einer Senke liegt, ist im Datenraum größer als im latenten Raum, in welchem die Peaks flachgedrückt sind. Bei Helix stimmen die Distanzen ebenfalls nicht überein, denn im eindimensionalen latenten Raum ist die Helix an einer Stelle aufgeschnitten, sodass die im Datenraum auf verschiedenen Seiten der Schnittstelle liegenden Punkte im latenten Raum sehr weit voneinander entfernt sind.

Für den experimentellen Vergleich werden pro realem und künstlichem Da-



(a) Swiss-roll

(b) Broken Swiss-roll

(c) Twin-peaks

(d) Helix

Abb. 3.1: Künstliche Datensätze (ohne HD)

tensatz 300 Muster zufällig ausgewählt. Besitzt ein Datensatz weniger Muster, wird der gesamte Datensatz verwendet. Mit 300 Mustern pro Datensatz ist die Abdeckung des Datenraum zwar eher gering, eine höhere Musteranzahl hätte jedoch eine zu lange Laufzeit der Experimente zur Folge gehabt.

3.1.3 Verwendete Implementierungen

Für den experimentellen Vergleich wurde mit Python[2] ein Grundgerüst implementiert, welches die Dimensionsreduktionsverfahren startet und die Me-

[2] https://www.python.org/ (letzter Zugriff am 16.11.2016).

triken für deren Resultate berechnet. Folgende Implementierungen wurden für die Dimensionsreduktionsverfahren verwendet: scikit-learn [Ped+11] für mMDS, nMDS, MLLE und HLLE, Matlab-Code[3] für EE nach der Beschreibung in [VCP12], Matlab-Toolbox[4] (Version 2b vom 05.04.2007) für UKR, Eigenimplementierung von UNN in Python nach [Kra15, Abschnitt 4.1], Python-Code für itUKR von dessen Autor Daniel Lückehe und für alle übrigen Verfahren die Matlab-Toolbox für Dimensionsreduktion [Maa]. Zur Implementierung von CE in der Matlab-Toolbox sei angemerkt, dass CE dort auch als *Curvilinear Component Analysis* (CCA) bezeichnet wird. CCA ist jedoch ein gänzlich anderes, auf einer SOM beruhendes Dimensionsreduktionsverfahren [DH95]. Den Kommentaren im Code zu CCA kann aber entnommen werden, dass CE anstatt CCA gemeint ist. Aus dieser Toolbox [Maa] wird ebenfalls die out-of-sample-Erweiterung für nichtparametrische auf Nachbarschaftsgraphen basierende Dimensionsreduktionsverfahren verwendet (siehe Abschnitt 2.5).

3.1.4 Einstellung der Parameter

Um die Dimensionsreduktionsverfahren sinnvoll miteinander vergleichen zu können, wurde für jedes Verfahren bei jedem Datensatz und jeder Metrik die Parametereinstellung bestimmt, bei welcher das Verfahren eine möglichst gute Einbettung liefert. Die Metrikwerte dieser Einbettung werden mit denen anderer Verfahren verglichen.

Dafür wurden zunächst angemessene Wertebereiche für die Parameter festgelegt, indem visualisierte Einbettungen von Swiss-roll ausgewertet und Empfehlungen der Autoren der Verfahren berücksichtigt wurden. Anschließend wurde für jedes Verfahren aus den Wertebereichen seiner Parameter eine Teilmenge an Werten ausgewählt (siehe Tabellen A.1 und A.2) und das Verfahren

[3] `http://faculty.ucmerced.edu/mcarreira-perpinan/software.html` (letzter Zugriff am 16.11.2016).
[4] `http://www.sklanke.de/` (letzter Zugriff am 16.11.2016).

mit jeder möglichen Kombination dieser Werte auf allen Datensätzen ausgeführt. Die Teilmenge der Werte wurde so gewählt, dass die Wertebereiche gleichmäßig abgedeckt waren und sich für jedes Verfahren pro Datensatz nahezu gleich viele Durchläufe (*Experimente*) ergaben. Jedes der Experimente wurde 25 Mal[5] wiederholt; die Werte der Metriken wurden über die Wiederholungen gemittelt. Als beste Parametereinstellung eines Verfahrens für einen bestimmten Datensatz und eine bestimmte Metrik wurde die des Experimentes gewählt, bei welchem der Mittelwert der Metrik der beste aller Experimente des Verfahrens auf diesem Datensatz war. Dieser beste Mittelwert geht in die Auswertung ein.

Die Metriken ENX, DSRE, CRR und EKNN besitzen ebenfalls einen einzustellenden Parameter, die Nachbarschaftsgröße k, dessen Wert für den experimentellen Vergleich auf 15 gesetzt wurde. Die Wahl orientiert sich an den Ergebnissen von [LV09]. In der Arbeit werden für verschiedene Verfahren die Werte unterschiedlicher Metriken als Funktionsgraphen in Abhängigkeit von k dargestellt. Für $0 < k \leq 10$ weisen diese teilweise starke Schwankungen der Metrikwerte auf, ab $k \geq 20$ meist einen eher stetigen Verlauf. Mit steigendem k unterscheiden sich die Metrikwerte zudem tendenziell immer weniger. Da in der vorliegenden Arbeit mit 300 Mustern die Datensätze deutlich kleiner als die in [LV09] verwendeten sind (1000 Muster oder mehr), wird für diese Arbeit $k = 15$ als sinnvoller Kompromiss zwischen der Glättung der Werte und der Aussagekraft der Metriken angesehen.

3.2 Vergleich der konvexen Verfahren

Insgesamt gehen pro Dimensionsreduktionsverfahren für jeden Datensatz sechs Metrikwerte in die Auswertung ein, zusammen $29 \cdot 13 \cdot 6 = 2262$ Werte. Um den Vergleich übersichtlich zu halten, werden die Ergebnisse konvexer und

[5] Die Experimente für FA auf den Datensätzen ORL und CNS und die für CE auf ORL wurden wegen zu langer Laufzeit nur drei Mal wiederholt.

nichtkonvexer Verfahren zunächst getrennt anhand ihrer Ränge miteinander verglichen. Dafür wird jedem Verfahren V pro Datensatz D bezüglich jeder Metrik M ein Rang $R_V(D, M)$ zugewiesen. Erreicht beispielsweise PCA bei Helix den drittbesten EKS-Wert aller konvexen und nichtkonvexen Verfahren, erhält es den Rang $R_{\text{PCA}}(\text{Helix}, \text{EKS}) = 3$; Verfahren mit gleichen Metrikwerten bekommen den gleichen Rang.

Für die Untersuchung des ersten Ziels der Evaluation, den Vergleich der verschiedenen Eigenschaften der Verfahren (siehe Abschnitt 3.1.1), werden für jedes Verfahren V dessen Ränge bezüglich einer Metrik M über alle 13 Datensätze gemittelt:[6]

$$\varnothing_{R_V}(M) = \frac{1}{13} \sum_{D \in \{\text{Swiss-roll},\ldots,\text{ORL}\}} R_V(D, M). \tag{3.12}$$

Diese Werte sind für die konvexen Verfahren in Abbildung 3.2 zu sehen; PCA nimmt beispielsweise im Mittel den Rang 10.8[7] bezüglich des ENX ein. Der Farbverlauf verdeutlich die Abstufung der Werte innerhalb einer Spalte: Kleine Werte werden gelb, große rot dargestellt.

Für das zweite Ziel, den Vergleich der Verfahren bei unterschiedlichen Datensätzen, wird für jedes Verfahren V pro Datensatz D sein durchschnittlicher Rang $\varnothing_{R_V}(D)$ durch Mittelung der Ränge bezüglich der sechs Metriken berechnet:

$$\varnothing_{R_V}(D) = \frac{1}{6} \sum_{M \in \{\text{ENX},\ldots,\text{EKNN}\}} R_V(D, M). \tag{3.13}$$

Für die konvexen Verfahren sind diese Werte in Abbildung 3.4 zu sehen.

Ähnlich wie in [FD+14] werden Ränge (bzw. deren Mittelwerte) anstatt absoluter Metrikwerte verwendet, da die Verfahren zunächst daraufhin unter-

[6] Metriken, die nicht bei allen Datensätzen berechnet werden können, werden bei den jeweiligen Datensätzen nicht in die Durchschnittsberechnung einbezogen.

[7] Das Dezimalkomma von Zahlen wird in dieser Arbeit durch einen Punkt dargestellt.

Verfahren	ENX	EKS	DSRE	E1NN	CRR	EKNN	Ø
PCA	**10.8**	7.0	12.0	16.9	15.4	11.5	12.3
Isomap	12.8	9.2	13.7	9.7	8.3	15.0	11.4
KPCA	19.8	20.0	19.9	15.2	15.5	17.0	17.9
MVU	11.8	6.4	12.0	14.0	11.4	**1.0**	**9.4**
DM	14.3	13.5	15.8	11.7	10.5	15.5	13.5
LLE	16.2	15.3	13.9	11.3	10.1	20.5	14.6
MLLE	**10.8**	15.9	**10.3**	10.9	7.6	10.0	10.9
NPE	20.9	18.0	20.6	18.0	17.8	5.0	16.7
LE	**10.8**	15.2	13.4	**6.5**	**7.5**	16.0	11.6
LPP	21.4	17.4	18.6	12.0	15.5	12.0	16.2
CE	12.6	**6.3**	12.7	12.3	9.5	11.0	10.7
HLLE	11.0	16.4	11.4	11.6	11.9	7.0	11.6
LTSA	13.2	15.8	12.9	12.1	9.9	12.5	12.7
LLTSA	15.6	13.0	13.0	11.1	15.7	16.0	14.1

Abb. 3.2: Durchschnittliche Ränge pro Metrik und im Mittel (∅-Spalte); kleine Werte sind gut (gelb hinterlegt), Verfahren unterschiedlicher Familien sind durch **- - -** getrennt

sucht werden, welche Verfahren besser als andere sind. Die Größe der Qualitätsunterschiede der Verfahren wird im gemeinsamen Vergleich der konvexen und nichtkonvexen Verfahren untersucht (Abschnitt 3.4).

Im Folgenden wird für die konvexen Verfahren zunächst das erste Evaluationsziel auf Basis der mit Gleichung 3.12 berechneten Werte (Abschnitt 3.2.1) und anschließend das zweite Ziel mit den sich aus Gleichung 3.13 ergebenden Werten untersucht (Abschnitt 3.2.2). Die Visualisierungen werden in Abschnitt 3.2.3 ausgewertet.

3.2.1 Vergleich hinsichtlich der Eigenschaften

Wird für jedes Verfahren der Mittelwert über die durchschnittlichen Ränge bezüglich der Metriken gebildet (die ∅-Spalte in Abbildung 3.2), besitzt MVU den besten durchschnittlichen Rang. KPCA und die linearen Verfahren NPE und LPP nehmen die schlechtesten Ränge ein; die Ränge der übrigen Verfahren mit CE und MLLE an der Spitze befinden sich zwischen diesen beiden Gruppen und unterscheiden sich untereinander nur wenig. Dass die

linearen Verfahren NPE, LPP und LLTSA schlechter als deren nichtlineare Äquivalente sind, ist zu erwarten, da in Datensätzen oftmals nichtlineare Information enthalten ist, die von nichtlinearen Verfahren besser erfasst werden kann. Aus diesem Grund ist es auffällig, dass KPCA als nichtlineare Variante von PCA schlechter als dieses ist.

Bei der Auswertung der übrigen Tabellenspalten fällt auf, dass die insgesamt schlechtesten Verfahren (KPCA, NPE, LPP) bei allen Metriken schlechte Ränge einnehmen, die insgesamt besten Verfahren (MVU, CE, MLLE) hingegen nicht bezüglich aller Metriken die besten Ränge besitzen. Beispielsweise MVU schneidet beim EKNN[8] und beim EKS sehr gut ab während MLLE beim ENX, DSRE und CRR sehr gute Ergebnisse hat. Außerdem nehmen Verfahren, die nicht zu den durchschnittlich besten gehören, bezüglich einiger Metriken teilweise ebenso gute Ränge ein. Beispielsweise LE erzielt beim CRR und E1NN die besten Ränge und PCA gehört bezüglich des ENX, des DSRE und des EKS zu den besten Verfahren. Dies zeigt, dass solche Verfahren bei Anwendungen, für die nur bestimmte Metriken wichtig sind, gegenüber den insgesamt besten Verfahren Alternativen darstellen können.

Ein Nachteil der durchschnittlichen Ränge ist, dass keine Aussage darüber möglich ist, ob die Ränge eines Verfahrens zwischen den Datensätzen großen Schwankungen unterliegen oder nahezu gleich bleiben. Ein differenzierteres Bild ergibt sich, wenn pro Metrik nicht der durchschnittliche Rang eines Verfahrens, sondern die Häufigkeit betrachtet wird, mit der der Rang des Verfahrens bezüglich dieser Metrik unter den besten beziehungsweise den schlechtesten Rängen aller Verfahren ist. Hinsichtlich der Ranghäufigkeiten ist ein Verfahren gut, wenn es oft unter den besten fünf und selten unter den schlechtesten zehn Verfahren ist,[9] also in Abbildung 3.3a einen hohen und in

[8] Dieser Wert darf nicht überinterpretiert werden, da der EKNN nur bei zwei Datensätzen Anwendung findet und MVU bei einem dieser Datensätze kein Ergebnis erzielt hat. Folglich wurde für MVU nur ein Ergebnis in die Durchschnittsberechnung einbezogen.

[9] Es wurden diese ungleichen Grenzen gewählt, um zum einen stärker zu erfassen, wenn der Rang eines Verfahren schlechter als mittelmäßig ist, und zum anderen nur wenige

Abbildung 3.3b einen niedrigen Wert erzielt.

Auffällig ist, dass PCA von allen Verfahren am häufigsten unter den fünf besten ist (siehe Σ-Spalte in Abbildung 3.3a). Da es allerdings genauso oft zu den letzten zehn gehört, schneidet es insgesamt bezüglich des durchschnittlichen Ranges nur mit mittelmäßigem Erfolg ab (siehe Abbildung 3.2); LE zeigt ein ähnliches Verhalten. Bei HLLE ist die Situation umgekehrt: Es gehört nur selten zu den schlechtesten Verfahren ist aber auch selten eines der besten. Es ist also festzustellen, dass ein gutes Abschneiden in der linken Tabelle nicht zwangsläufig ein gutes Resultat in der rechten und umgekehrt bedeutet. Neben PCA nehmen auch CE, LE und Isomap häufig Ränge unter den besten fünf ein; MVU und MLLE befinden sich im mittleren Bereich. Auffallend selten unter den besten fünf Verfahren sind NPE, LPP und KPCA, die auch bei den durchschnittlichen Rängen (Abbildung 3.2) schlecht abschneiden. Auffallend ist in Abbildung 3.3a ebenfalls, dass beim EKNN nur wenige Verfahren einen der ersten fünf Ränge erreichen. Die Ursache hierfür ist, dass der EKNN nur bei den zwei Datensätzen mit kontinuierlichen Labels (Boston und RCT) anwendbar ist. Folglich können beim EKNN maximal $2 \cdot 5$ Verfahren unter den besten fünf sein. Am seltensten unter den schlechtesten zehn Verfahren sind Isomap, MVU, MLLE, HLLE und CE, am häufigsten KPCA, PCA, LPP und NPE.

Unter Betrachtung der Abbildungen 3.3a und 3.3b schneiden Isomap, MVU, CE und MLLE am besten ab, während bei den durchschnittlichen Rängen (Abbildung 3.2) MVU, CE und MLLE die besten sind. Da der durchschnittliche Rang von Isomap nur geringfügig schlechter als der von CE ist, kann auch Isomap neben MVU, CE und MLLE zu den besten konvexen Verfahren gezählt werden. Die schlechtesten Verfahren sind KPCA, LPP und NPE. PCA wird dieser Gruppe nicht zugeordnet, obwohl es am zweithäufigsten unter den schlechtesten zehn Verfahren ist, da dessen durchschnittlicher Rang

Verfahren unter die besten aufzunehmen.

Verfahren	ENX	EKS	DSRE	E1NN	CRR	EKNN	Σ
PCA	**6**	5	**6**	1	1	**1**	**20**
Isomap	1	2	0	5	**5**	0	13
KPCA	0	2	0	1	1	0	4
MVU	2	5	1	1	1	**1**	11
DM	0	1	0	2	4	0	7
LLE	0	1	0	3	3	0	7
MLLE	1	1	1	2	4	**1**	10
NPE	0	0	0	0	0	**1**	1
LE	2	2	0	**6**	4	0	14
LPP	0	0	0	2	0	0	2
CE	3	**6**	2	2	3	0	16
HLLE	0	1	1	1	2	0	5
LTSA	0	1	1	2	3	0	7
LLTSA	1	1	3	3	0	0	8

(a) Häufigkeit Rang $\leq$ 5; große Werte sind gut (rot hinterlegt)

Verfahren	ENX	EKS	DSRE	E1NN	CRR	EKNN	Σ
PCA	5	1	4	5	5	**0**	20
Isomap	1	**0**	1	2	**0**	1	**5**
KPCA	9	9	7	4	5	1	35
MVU	2	**0**	1	2	1	**0**	6
DM	1	3	3	1	3	**0**	11
LLE	4	2	3	1	1	1	12
MLLE	**0**	4	**0**	1	1	**0**	6
NPE	3	3	4	2	1	**0**	13
LE	1	4	2	1	1	1	10
LPP	4	3	3	1	2	**0**	13
CE	4	**0**	2	2	**0**	**0**	8
HLLE	**0**	5	1	**0**	1	**0**	7
LTSA	1	5	2	2	1	**0**	11
LLTSA	1	2	2	2	3	**0**	10

(b) Häufigkeit Rang $\geq$ 20; kleine Werte sind gut (gelb hinterlegt)

Abb. 3.3: Anzahl der Datensätze, bei denen ein Verfahren einen Rang ≤ 5 bzw. ≥ 20 bezüglich einer Metrik erreicht

deutlich besser ist. Da kein Verfahren in allen Metriken die besten Ränge erreicht, kann PCA eine Alternative darstellen, wenn insbesondere der ENX, DSRE und EKS (also die Struktur der Mannigfaltigkeit) optimiert werden sollen, und LE, wenn der CRR (die Informationserhaltung) und der E1NN im Vordergrund stehen.

3.2.2 Vergleich hinsichtlich der Datensätze

In diesem Abschnitt wird untersucht, ob die Qualität der Verfahren von den Datensätzen abhängt (zweites Ziel der Evaluation). Dazu werden, wie in Gleichung 3.13 beschrieben, die Ränge der Verfahren pro Datensatz über die sechs Metriken gemittelt; Abbildung 3.4 zeigt diese Werte $\varnothing_{R_V}(D)$. Sie wurden für jedes Verfahren getrennt nach künstlichen und realen Datensätzen gemittelt ($\varnothing$ (k.)- und $\varnothing$ (r.)-Spalten) und der Betrag der Abweichung zwischen beiden Mittelwerten (Abw.-Spalte) berechnet. Ein Eintrag ### bedeutet, dass das Verfahren aufgrund eines Fehlers zur Laufzeit keine Einbettung berechnen

Verfahren	Swiss	Broken	Helix	Twin	HD	Iris	Boston	Digits	RCT	MNIST	HIVA	CNS	ORL	Ø(k.)	Ø(r.)	Abw.
PCA	21.4	20.6	23.0	11.2	12.4	5.6	15.3	**4.2**	**3.8**	18.6	7.8	**6.2**	7.0	17.7	8.6	9.2
Isomap	8.6	**4.8**	9.6	6.6	18.4	12.0	13.3	14.8	8.8	14.0	8.0	7.0	17.0	**9.6**	11.9	2.3
KPCA	24.6	23.0	12.6	17.4	8.8	9.8	28.3	20.8	10.5	25.0	14.2	17.0	25.6	17.3	18.9	1.6
MVU	14.2	18.4	16.2	6.6	8.6	**4.6**	8.8	###	###	11.4	###	###	8.2	12.8	8.2	4.6
DM	14.2	16.0	12.8	14.8	10.2	13.8	15.3	8.4	14.0	13.0	13.0	8.4	20.4	13.6	13.3	0.3
LLE	13.2	16.6	13.8	18.8	12.2	14.0	19.3	13.0	15.5	8.6	11.2	11.0	13.2	14.9	13.2	1.7
MLLE	10.6	6.4	8.4	12.4	11.0	17.2	10.3	7.6	17.3	10.2	7.2	14.8	13.2	9.8	12.2	2.5
NPE	17.8	17.0	19.2	20.8	19.4	20.6	15.5	###	###	###	###	###	###	18.8	18.1	0.8
LE	**8.2**	11.4	**8.2**	12.8	17.2	16.2	21.0	8.4	9.0	**5.2**	**6.6**	10.4	11.0	11.6	11.0	0.6
LPP	17.8	18.0	16.4	20.4	12.6	15.4	19.0	###	###	###	###	###	###	17.0	17.2	**0.2**
CE	15.6	16.8	19.6	5.6	**7.2**	4.8	13.8	###	###	7.4	###	###	**5.8**	13.0	**7.9**	5.0
HLLE	11.8	11.4	**8.2**	12.0	17.0	12.8	**8.5**	###	###	13.4	###	###	15.6	12.1	12.6	0.5
LTSA	11.8	7.0	**8.2**	13.6	11.2	21.6	9.5	9.8	16.8	13.4	15.6	13.6	15.6	10.4	14.5	4.1
LLTSA	14.4	18.6	14.0	**5.0**	13.8	6.2	16.3	###	###	###	###	###	22.4	13.2	15.0	1.8

Abb. 3.4: Durchschnittliche Ränge pro Datensatz, Mittelwerte getrennt nach künstlichen (∅(k.)) und realen Datensätzen (∅(r.)) sowie Abweichung (Abw.) zwischen ∅(k.) und ∅(r.); kleine Werte sind gut (gelb hinterlegt)

konnte oder das Verfahren nach zu langer Laufzeit[10] aktiv abgebrochen wurde. In die Mittelwerte ∅ (k.) und ∅ (r.) gehen nur die Datensätze ein, bei denen ein Ergebnis vorliegt.

Die ∅ (k.)- und ∅ (r.)-Spalten zeigen, dass Isomap, MLLE und LTSA bei den künstlichen Datensätzen und CE, MVU und PCA bei den realen Datensätzen im Durchschnitt die besten Ränge erlangen. Da CE und MVU jedoch auf den realen Datensätzen häufig kein Ergebnis liefern, werden die nächstbesten fehlerfreien Verfahren LE und Isomap als mindestens gleich gut angesehen. Die schlechtesten Verfahren sind bei den künstlichen Datensätzen NPE, PCA, KPCA und LPP, bei den realen sind es mit Ausnahme von PCA dieselben. Dies überrascht nicht, da KPCA, LPP und NPE auch hinsichtlich der durchschnittlichen Ränge pro Metrik (Abschnitt 3.2.1) am schlechtesten abschneiden.

[10] Da MVU und CE bei den Datensätzen Digits und HIVA nach vierwöchiger Laufzeit noch kein Ergebnis für das erste Experiment geliefert hatten, wurden sie abgebrochen, um die Auswertung des experimentellen Vergleichs vornehmen zu können.

Die Abweichungen zwischen den Mittelwerten der realen und künstlichen Datensätze (Abw.-Spalte) sind bei PCA fast doppelt so groß wie bei der nächstgrößten Abweichung eines Verfahren; PCA erzielt bei realen Datensätzen deutlich bessere Ränge als bei künstlichen. Dies stimmt mit den Ergebnissen von [MPH09] überein und erklärt den Befund aus Abschnitt 3.2.1, dass PCA häufig unter den besten aber auch oft unter den schlechtesten Verfahren ist. Bei realen Datensätzen ebenfalls sichtbar besser sind MVU und CE. LTSA besitzt eine ähnlich hohe Abweichung wie diese, ist aber bei künstlichen Datensätzen erfolgreicher. Die Abweichungen der Mittelwerte der übrigen Verfahren sind wesentlich kleiner. Insgesamt sind genauso viele Verfahren auf künstlichen Datensätzen besser wie auf realen. Die Einzelergebnisse für die künstlichen und realen Datensätze werden im Folgenden näher betrachtet.

Die meisten Verfahren erzielen bei Broken Swiss-roll einen schlechteren durchschnittlichen Rang als bei Swiss-roll, was zeigt, dass die Einbettung nichtzusammenhängender Mannigfaltigkeiten für konvexe Verfahren schwieriger ist. Beim Vergleich der Ergebnisse bei Helix und Twin-peaks (beide besitzen Mannigfaltigkeiten, die nicht isometrisch zum euklidischen Raum sind) fällt auf, dass die Ränge bei Helix kleiner sind, oft sogar besser als bei Swiss-roll. Ein möglicher Grund dafür ist die geringere intrinsische Dimensionalität von Helix.

Abbildung 3.4 zeigt deutlich, dass Probleme ausschließlich bei der Einbettung realer Datensätze auftreten. Bei Iris und Boston gibt es keine Fehler, da sowohl deren Datenraum- als auch deren intrinsische Dimensionalität sehr gering sind. Bei MNIST und ORL treten aufgrund ihrer geringen intrinsischen Dimensionalität ebenfalls weniger Fehler auf. Auffällig ist, dass bis auf MVU nur teilspektrale Verfahren von Fehlern betroffen sind, insbesondere lineare (NPE, LPP, LLTSA) und semidefinite Programmierung nutzende Verfahren (CE, MVU). Außerdem kann HLLE teilweise keine Einbettung berechnen. Ursache für die Probleme von NPE und LPP ist, dass die Datensätze zu wenig Muster bei zu hoher Datenraumdimensionalität beinhalten.

HLLE liefert keine Ergebnisse, da die intrinsische Dimensionalität größer als die Nachbarschaftsgröße ist. Eine weitere häufige Ursache sind unlösbare Eigenwertprobleme (MVU, CE, LLTSA). Das Lösen der Eigenwertprobleme der teilspektralen Verfahren ist laut [MPH09] schwierig, da die kleinsten Eigenwerte (ohne Null) gesucht werden und sich diese oft nur geringfügig vom Wert Null unterscheiden. Die dritte Ursache fehlender Einbettungen sind zu lange Ausführungsdauern (MVU und CE bei Digits und HIVA). Diese lassen sich auf die Verwendung semidefiniter Programmierung zurückführen [MPH09].

Die Ergebnisse dieses Abschnitts zeigen zusammen mit den Erkenntnissen aus Abschnitt 3.2.1, dass MVU, CE, MLLE und Isomap die besten durchschnittlichen Ränge erreichen. Wegen der Fehleranfälligkeit von CE und MVU auf realen Datensätzen werden PCA und LE als besser angesehen, da diese fehlerfrei sind und ebenfalls gute Ränge besitzen. Am schlechtesten schneiden insgesamt KPCA, NPE und LPP ab. Die Ergebnisse der voll- und teilspektralen Verfahren unterscheiden sich nicht wesentlich, außer dadurch, dass bei teilspektralen deutlich häufiger bei realen Datensätzen keine Einbettung möglich ist. Die Hauptursache dafür ist die hohe intrinsische Dimensionalität der Datensätze.

3.2.3 Vergleich der Visualisierungen

In diesem Abschnitt werden die visualisierten Einbettungen der konvexen Verfahren für künstliche (Tabelle 3.2) und reale Datensätze (Tabelle 3.3) gezeigt, deren Mannigfaltigkeit visualisierbar, d. h. ein-, zwei- oder dreidimensional, ist. Es wird die Einbettung der Parameterbelegung visualisiert, bei der das Verfahren den besten EKS-Wert erreicht. Diese Einbettungen scheinen bei einigen, willkürlich ausgewählten Verfahren (u. a. Isomap, EE, CE) für die Visualisierung geeigneter zu sein, als Einbettungen, die bezüglich anderer Metriken besser sind. Es wird jeweils die Einbettung des ersten Durchlaufs des Experiments gezeigt.

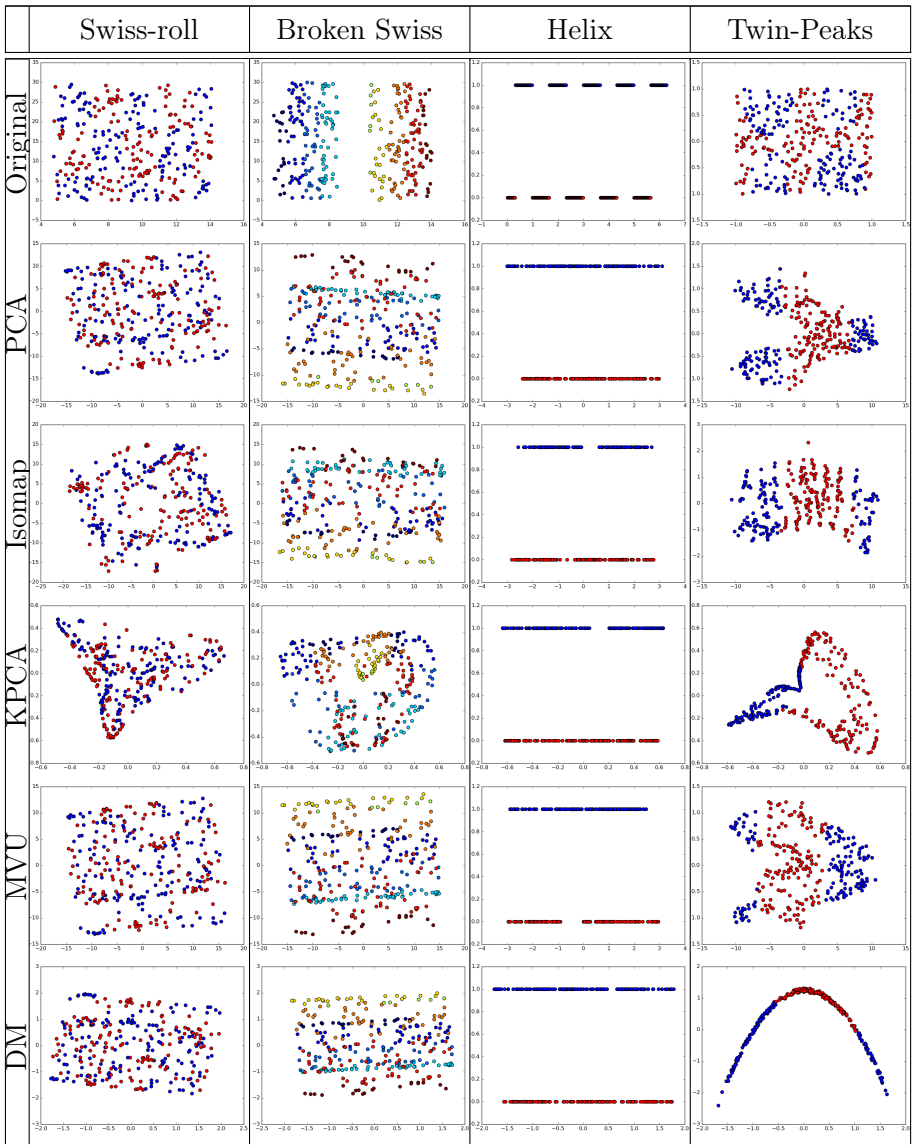
Die Visualisierungen[11] werden danach untersucht, wie gut die einzelnen Klassen getrennt sind, ob die Mannigfaltigkeit Löcher oder Häufungen von Punkten enthält und ob Ähnlichkeiten verschiedener Verfahren zu sehen sind. Bei künstlichen Mannigfaltigkeiten werden die Einbettungen zusätzlich mit der originalen Mannigfaltigkeit verglichen.

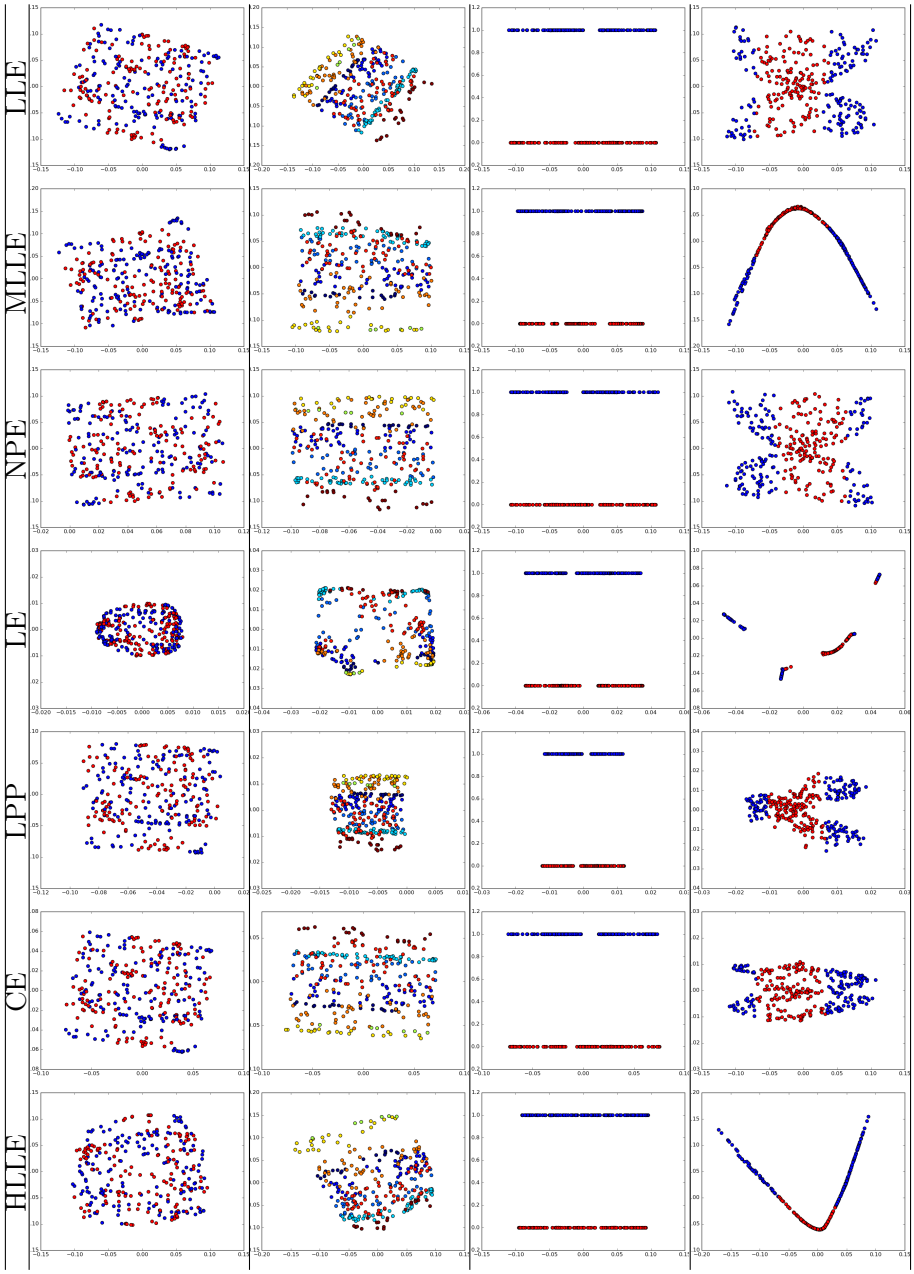
Zur Mannigfaltigkeit von Helix sei angemerkt, dass diese eindimensional ist, aber zweidimensional abgebildet wird, damit die Wechsel zwischen Punkten verschiedener Farben besser erkennbar sind.

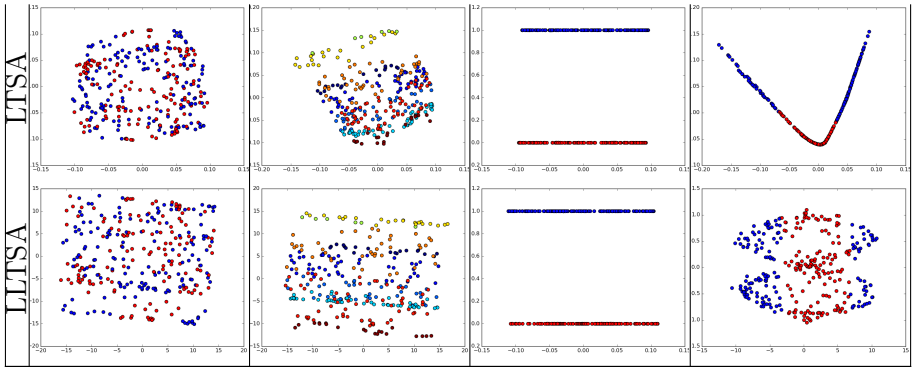
Künstliche Datensätze

Bei Swiss-roll ist das Schachbrettmuster in keiner Einbettung gut zu erkennen. Dies ist jedoch nicht primär auf ein Versagen der Verfahren sondern auf eine zu geringe Anzahl Muster zurückzuführen, denn auch in der originalen Mannigfaltigkeit ist das Muster nicht gut sichtbar. Die Rechteckstruktur ist dagegen bei allen Verfahren außer KPCA und LE gut erhalten; bei Isomap, HLLE und LTSA ist in der Mitte die Dichte der Punkte eher gering. Es fällt auf, dass die Einbettungen für Swiss-roll von MVU, PCA, DM, LLE und CE bis auf Drehungen nahezu identisch sind. Die Einbettung von Broken Swiss-roll stimmt bei keinem Verfahren mit dem Original überein: Alle Verfahren bis auf KPCA und LE erzeugen eine ähnliche, falsche Mannigfaltigkeit. Die Ergebnisse bei Helix sind ebenfalls schlecht: Keine der Mannigfaltigkeiten zeigt die wechselweise Abfolge der Farben. Bei Twin-peaks hingegen sind Punkte unterschiedlicher Farben gut getrennt, wenn auch die Struktur nicht erhalten bleibt. Bei LLE, NPE und LLTSA ist diese vergleichsweise gut, jedoch in der Mitte „zusammengeknotet“. KPCA und LE weisen wie bei Broken Swiss-roll und Swiss-roll Löcher in der Mannigfaltigkeit auf und HLLE und LTSA haben hier wie auch bei den anderen Datensätzen exakt die gleiche Einbettung.

[11] Die Grafiken der Einbettungen wurden mit Matplotlib erzeugt: http://matplotlib.org/ (letzer Zugriff am 16.11.2016).


Swiss-roll
Broken Swiss
Helix
Twin-Peaks
Original
PCA
Isomap
KPCA
MVU
DM


HLLE
CE
LPP
LE
NPE
MLLE
LLE



Tab. 3.2: Visualisierte Einbettungen konvexer Verfahren (künstliche Datensätze)

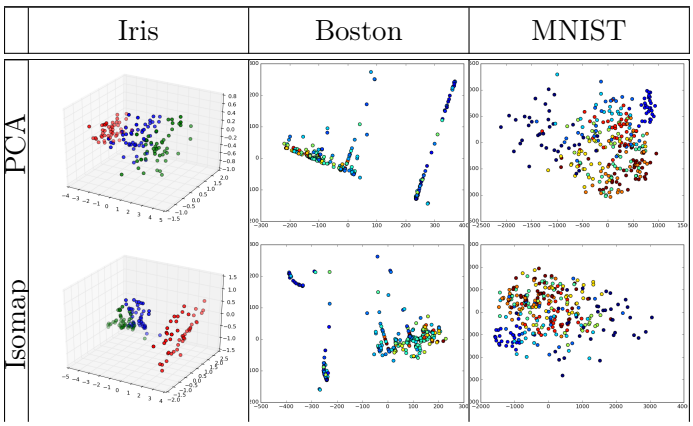
Reale Datensätze

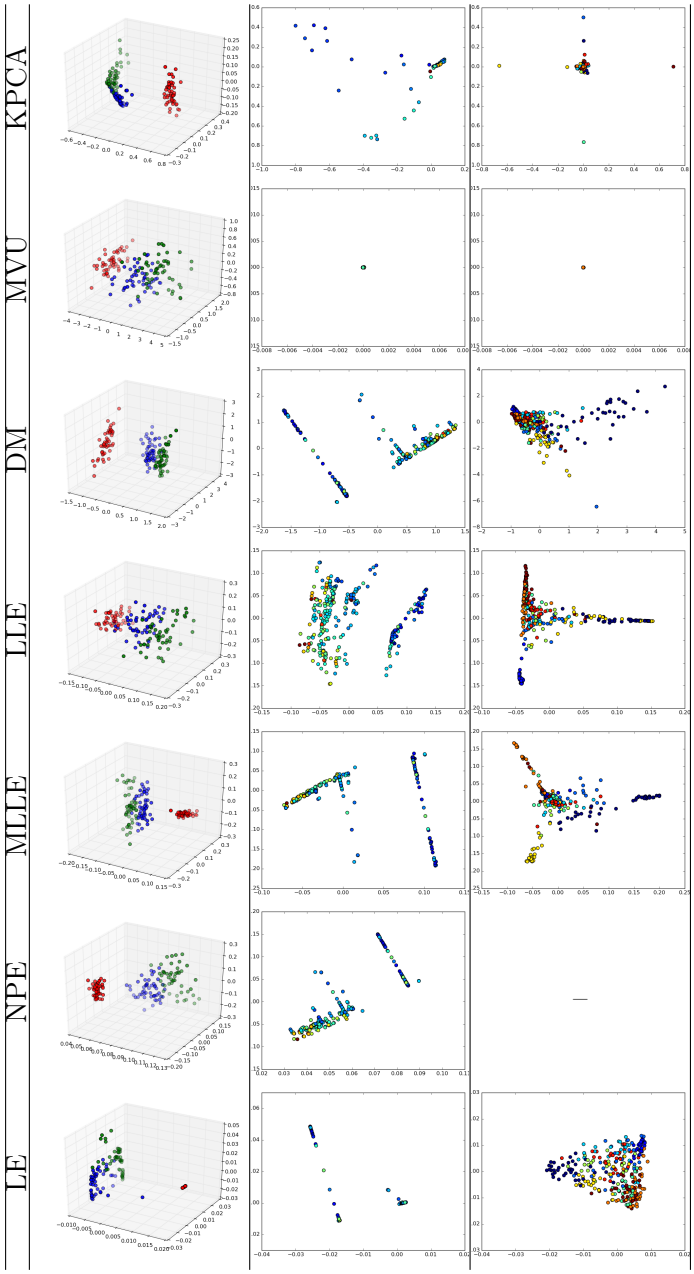
Bei realen Datensätzen gibt es aufgrund der unbekannten originalen Mannigfaltigkeiten wenig Anhaltspunkte für eine gute Visualisierung. Bei Datensätzen mit Klassen können jedoch Mannigfaltigkeiten mit räumlich getrennten Gruppen von latenten Punkten, die derselben Klasse angehören, als sinnvoll angesehen werden, denn in der praktischen Anwendung verfügt ein Datensatz in der Regel nicht über Labels, sodass der Betrachter die Daten allein auf Grundlage der räumlichen Strukturen interpretieren muss.

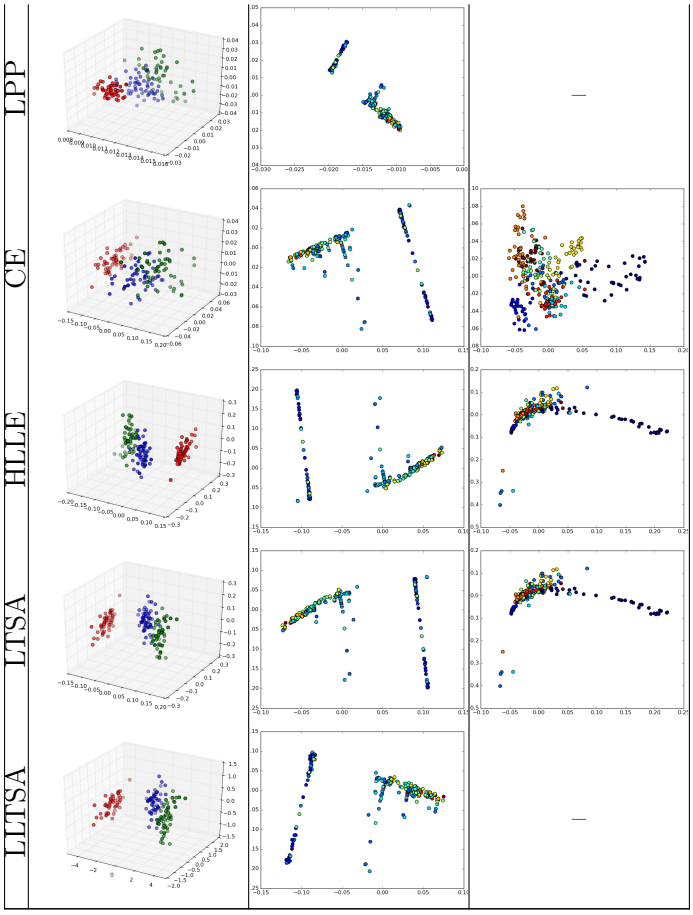
Bei Iris sind die latenten Punkte unterschiedlicher Klassen selten durchmischt, oft sind die roten Punkte sogar räumlich von den anderen getrennt. Die beste Trennung, sodass auch die grünen und blauen Punkte in einzelne Punktewolken unterschieden werden können, erreichen DM, MLLE, HLLE und LTSA; in dieser Hinsicht sind PCA, MVU, LPP und CE schlecht. Bei Boston ist die Qualität der Mannigfaltigkeiten schwieriger zu beurteilen, da keine Klassen, sondern kontinuierliche Labels vorhanden sind und somit nicht unbedingt Gruppen von Punkten zu erwarten sind. Dennoch ist bei den

meisten Verfahren eine Linie dunklerer Punkte räumlich getrennt von einer Ansammlung hellerer Punkte, von der aus andere Punktlinien senkrecht abzweigen; MLLE, LTSA und HLLE erzeugen die gleiche Visualisierung. LLE verteilt die Punkte dagegen mehr, trennt aber trotzdem die dunklen und hellen Punkte voneinander. Obwohl MNIST über Klassen verfügt, sind bei keiner Mannigfaltigkeit räumlich getrennte Gruppen gleichfarbiger Punkte erkennbar und auch innerhalb der Mannigfaltigkeiten sind Punkte unterschiedlicher Farbe stark durchmischt. Lediglich bei LE und MLLE lassen sich an den Spitzen der dreiecksförmigen Einbettung gehäuft Punkte gleicher Farbe erkennen. Zusätzlich werden die Punkte in vielen Einbettungen gedrängt angeordnet, MVU und KPCA häufen die Punkte sogar.

Insgesamt sind die visualisierten Einbettungen von Iris gegenüber den anderen realen Datensätzen vergleichsweise gut. Ein Grund für die gegenüber Iris schlechtere Trennung der Klassen bei MNIST könnte die höhere Datenraumdimensionalität von MNIST (784 statt 4) sein. Die visualisierten realen und künstlichen Mannigfaltigkeiten sind nicht gut genug und unterscheiden sich nicht ausreichend deutlich, um ein oder mehrere konvexe Verfahren als für Visualisierungszwecke am geeignetesten einzustufen.




KPCA
MVU
DM
LLE
MLLE
NPE
LE



Tab. 3.3: Visualisierte Einbettungen konvexer Verfahren (reale Datensätze)

3.2.4 Zusammenfassung

Der Vergleich der konvexen Verfahren zeigt, dass unter Berücksichtigung der durchschnittlichen Ränge der Verfahren pro Metrik und Datensatz MLLE, Isomap, PCA und LE die besten konvexen Verfahren sind. Es können kei-

Verfahren	ENX	EKS	DSRE	E1NN	CRR	EKNN	Ø
mMDS	7.4	2.7	8.2	11.5	14.1	23.0	11.1
SM	7.2	**1.7**	7.0	14.3	14.8	5.5	8.4
nMDS	25.8	22.2	25.8	24.5	22.9	25.5	24.4
FA	17.7	12.5	19.3	18.8	14.8	15.5	16.4
PPCA	17.8	16.0	16.2	16.6	19.3	19.5	17.6
GPLVM	10.8	6.6	11.4	15.3	14.7	10.0	11.5
LLC	15.1	14.9	12.8	15.5	14.3	17.0	14.9
MC	17.5	18.6	20.0	20.2	17.4	23.5	19.6
SNE	8.5	15.4	11.7	13.2	13.4	13.5	12.6
t-SNE	**4.9**	17.2	8.2	5.4	6.1	5.0	7.8
EE	5.1	8.5	7.1	**4.2**	**4.0**	11.0	**6.6**
UKR	12.2	15.9	**3.0**	9.0	6.5	**1.5**	8.0
UNN	15.5	20.1	14.4	13.3	13.2	13.0	14.9
itUKR	15.6	17.5	15.6	13.3	13.3	8.0	13.9
Autoenc.	21.5	17.5	22.1	21.0	18.6	19.5	20.0

Abb. 3.5: Durchschnittliche Ränge pro Metrik und im Mittel (∅-Spalte); kleine Werte sind gut (gelb hinterlegt)

ne Qualitätsunterschiede zwischen den Familien erkannt werden, außer dass teilspektrale Verfahren gegenüber hohen intrinsischen Dimensionalitäten realer Datensätze weniger robust sind. Zudem geben die Visualisierungen meist keine wesentliche Hilfestellung bei der Einschätzung der Qualität der Einbettungen.

3.3 Vergleich der nichtkonvexen Verfahren

Der Vergleich der nichtkonvexen Verfahren ist genauso aufgebaut wie der der konvexen Verfahren. Zuerst werden die durchschnittlichen Ränge der Verfahren pro Metrik $\varnothing_{R_V}(M)$ (Abschnitt 3.3.1) und pro Datensatz $\varnothing_{R_V}(D)$ (Abschnitt 3.3.2) und anschließend die Visualisierungen (Abschnitt 3.3.3) untersucht.

3.3.1 Vergleich hinsichtlich der Eigenschaften

Bei den in Abbildung 3.5 gezeigten durchschnittlichen Rängen pro Metrik $\varnothing_{R_V}(M)$ (Gleichung 3.12) heben sich im Gegensatz zur entsprechenden Ab-

bildung der konvexen Verfahren (Abbildung 3.2) gute und schlechte Verfahren auffallend deutlich voneinander ab: Gute Verfahren erreichen bei fast allen Metriken gute Ränge, schlechte Verfahren hohe. Zudem ist ein Zusammenhang zwischen dem durchschnittlichen Rang und der Familienzugehörigkeit erkennbar. Besonders gut schneiden die SNE-Varianten EE und t-SNE, UKR und die MDS-Varianten mMDS und SM ab. Die probabilistischen Verfahren, die ein LVM oder ein MM nutzen, die Regressionsverfahren mit stochastischer Erzeugung möglicher latenter Punkte (UNN, itUKR) und der Autoencoder schneiden deutlich schlechter ab. GPLVM ist jedoch nur geringfügig schlechter als mMDS. Überraschend ist, dass nMDS einen deutlich höheren durchschnittlichen Rang als mMDS besitzt, obwohl es sich von mMDS nur darin unterscheidet, dass es statt der absoluten Distanzen deren Ränge bewahrt. Im Vergleich zu den konvexen Verfahren, deren durchschnittlichen Ränge (∅-Spalte) sich nur zwischen 9 und 16 bewegen, unterscheiden sich die der nichtkonvexen Verfahren mit einer Spannweite von 6 bis 24 deutlich stärker. Die übrigen Spalten zeigen, dass, anders als bei den konvexen Verfahren, außer den insgesamt besten Verfahren (EE, t-SNE, UKR, SM, mMDS) keine anderen Verfahren Alternativen für spezielle Metriken darstellen. Ungewöhnlich ist, dass EE und t-SNE jeweils bei fünf von sechs Metriken sehr gute Ränge erreichen.

Im Gegensatz zu den konvexen Verfahren sind die Werte der Tabellen 3.6a und 3.6b insofern konsistent, als Verfahren, die selten unter den besten sind (Σ-Spalte), häufig zu den schlechtesten gehören und umgekehrt. Eine Ausnahme bildet LLC, welches in der rechten Abbildung im Vergleich zu seinem Wert in der linken Abbildung und dem von GPLVM besser als erwartet abschneidet. EE, UKR, t-SNE, mMDS und SM können wegen ihres häufigen Vorkommens unter den besten und wegen ihres seltenen Vorkommens unter den schlechtesten Verfahren als beste Verfahren bezeichnet werden, wie auch die durchschnittlichen Ränge zeigen; EE ist mit Abstand das beste Verfahren. nMDS und der Autoencoder schneiden insgesamt am schlechtesten ab.

Verfahren	ENX	EKS	DSRE	E1NN	CRR	EKNN	Σ
mMDS	8	**12**	4	1	1	0	26
SM	7	11	6	0	1	1	26
nMDS	0	0	0	0	1	0	1
FA	0	1	0	0	2	0	3
PPCA	1	1	0	1	1	0	4
GPLVM	5	5	6	2	1	1	20
LLC	0	0	2	0	1	0	3
MC	0	0	0	0	1	0	1
SNE	6	0	4	2	3	1	16
t-SNE	7	1	7	7	8	1	31
EE	**9**	5	7	**8**	**10**	0	**39**
UKR	4	2	**12**	6	7	**2**	33
UNN	1	0	3	2	2	0	8
itUKR	1	0	3	1	3	0	8
Autoenc.	0	0	0	1	1	0	2

(a) Häufigkeit Rang $\leq$ 5; große Werte sind gut (rot hinterlegt)

Verfahren	ENX	EKS	DSRE	E1NN	CRR	EKNN	Σ
mMDS	1	**0**	1	1	2	1	6
SM	1	**0**	1	1	3	**0**	6
nMDS	12	9	12	9	9	2	53
FA	5	2	6	5	4	1	23
PPCA	6	5	4	4	5	1	25
GPLVM	5	1	3	3	4	**0**	16
LLC	3	2	2	1	3	**0**	11
MC	3	4	6	5	4	2	24
SNE	3	4	4	3	3	1	18
t-SNE	**0**	5	1	**0**	1	**0**	7
EE	**0**	1	**0**	**0**	**0**	**0**	**1**
UKR	2	5	1	2	1	**0**	11
UNN	4	6	3	3	3	**0**	19
itUKR	4	4	5	4	2	**0**	19
Autoenc.	9	5	10	9	7	1	41

(b) Häufigkeit Rang $\geq$ 20; kleine Werte sind gut (gelb hinterlegt)

Abb. 3.6: Anzahl der Datensätze, bei denen ein Verfahren einen Rang ≤ 5 bzw. ≥ 20 bezüglich einer Metrik erreicht

Außerdem zeigt Abbildung 3.6 wie auch Abbildung 3.5, dass bei t-SNE und UKR Schwierigkeiten bestehen, die Muster bezüglich des EKS korrekt einzubetten, da in der rechten Abbildung die Ränge beim EKS auffallend hoch sind. mMDS und SM schneiden beim E1NN und CRR weniger gut ab als beim ENX, EKS und DSRE, mMDS hat zusätzlich beim EKNN schlechte Ergebnisse. Bei EE hingegen ist bezüglich keiner Metrik eine Schwäche zu erkennen

Ungewöhnlich ist, dass alle nichtkonvexen Verfahren und auch alle bis auf drei konvexe Verfahren bezüglich des CRR mindestens ein Mal unter den besten fünf sind. Dies kommt dadurch zustande, dass im 2-Klassen-Datensatz HIVA mehr als 96% der Muster der gleichen Klasse angehören und somit selbst Einbettungen, die bezüglich anderer Metriken schlecht sind, einen guten CRR erzielen können, sodass viele Verfahren den Rang eins erhalten. Außerdem sei angemerkt, dass die sehr guten Ränge, die mMDS und SM beim EKS und UKR beim DSRE erzielen, dadurch zustande kommen, dass

Verfahren	Swiss	Broken	Helix	Twin	HD	Iris	Boston	Digits	RCT	MNIST	HIVA	CNS	ORL	Ø(k.)	Ø(r.)	Abw.
mMDS	9.0	11.6	20.2	**5.4**	8.6	8.0	11.3	5.2	4.3	11.2	5.2	6.4	6.0	11.0	7.2	3.8
SM	9.0	11.0	21.8	7.8	**8.2**	###	**5.5**	4.2	**1.8**	12.0	5.0	###	4.8	11.6	**5.5**	6.0
nMDS	29.0	29.0	29.0	28.8	26.8	28.0	28.5	20.4	22.8	21.0	12.8	17.8	22.6	28.5	21.7	6.8
FA	24.0	23.4	24.8	14.0	15.6	10.8	17.3	###	10.3	###	###	9.0	15.2	20.4	12.5	7.9
PPCA	20.8	19.2	18.4	27.8	19.6	24.8	18.0	11.8	21.3	16.0	8.8	7.2	11.0	21.2	14.9	6.3
GPLVM	21.0	20.2	23.0	11.0	11.0	5.6	14.5	**3.8**	3.3	17.4	4.6	6.2	7.0	17.2	7.8	9.4
LLC	18.4	11.6	14.8	16.4	14.6	18.2	11.3	12.0	19.0	14.4	10.4	17.0	11.4	15.2	14.2	1.0
MC	18.6	17.2	19.2	19.6	21.2	24.4	21.5	16.2	19.3	16.4	15.4	###	###	19.2	18.9	**0.3**
SNE	7.2	5.0	12.6	10.6	11.2	**5.4**	20.5	14.8	8.3	20.8	5.2	17.8	22.8	9.3	14.4	5.1
t-SNE	6.2	5.4	7.0	9.4	18.8	11.2	6.8	9.6	7.3	**1.8**	11.8	6.8	7.0	9.4	7.8	1.6
EE	6.0	**4.6**	**4.0**	10.8	9.2	6.8	8.5	5.0	11.3	2.8	**4.2**	**5.2**	**1.8**	**6.9**	5.7	1.2
UKR	**5.8**	7.0	7.8	13.4	21.4	16.4	8.8	7.6	8.3	2.8	6.6	7.2	6.2	11.1	8.0	3.1
UNN	13.6	18.2	9.8	23.2	23.6	17.6	13.8	16.8	15.0	9.6	12.0	10.6	15.4	17.7	13.8	3.8
itUKR	11.2	15.2	5.6	22.8	23.0	22.2	8.8	16.2	10.5	15.4	11.0	11.0	19.4	15.6	14.3	1.3
Autoenc.	27.2	28.0	24.2	18.8	16.2	18.6	25.3	20.2	13.5	21.4	8.2	16.6	23.2	22.9	18.4	4.5

Abb. 3.7: Durchschnittliche Ränge pro Datensatz, Mittelwerte getrennt nach künstlichen (∅(k.)) und realen Datensätzen (∅(r.)) sowie Abweichung zwischen ∅(k.) und ∅(r.); kleine Werte sind gut (gelb hinterlegt)

der EKS nahezu den Zielfunktionen von mMDS und SM entspricht und der DSRE die Zielfunktion von UKR ist. UNN und UKR optimieren ebenfalls den DSRE, jedoch nicht den originalen, sondern eine Abwandlung für einzelne latente Punkte (siehe Gleichung 2.73), sodass sie nur mittelmäßige Ränge beim DSRE erreichen. Unerklärlich ist allerdings, dass der Autoencoder auch beim DSRE schlecht abschneidet, obwohl dessen Trainigsziel äquivalent zur Minimierung des DSRE ist.

Das Ergebnis der Untersuchungen dieses Abschnitts ist, dass EE, t-SNE, UKR, SM und mMDS die besten nichtkonvexen Verfahren sind, wobei EE sich besonders dadurch hervorhebt, dass es bei allen Metriken gute Ränge erreicht, während t-SNE und UKR beim EKS, SM beim E1NN und CRR und mMDS zusätzlich beim EKNN schlechtere Ränge besitzen.

3.3.2 Vergleich hinsichtlich der Datensätze

Die Abbildung 3.7 enthält für jedes nichtkonvexe Verfahren pro Datensatz die über die Metriken gemittelten Ränge $\varnothing_{R_V}(D)$ (Gleichung 3.13). Aus der $\varnothing$ (k.)-Spalte ist abzulesen, dass EE mit Abstand den besten durchschnittlichen Rang auf den künstlichen Datensätzen erreicht. Es folgen t-SNE und SNE und mit Abstand mMDS, UKR und SM. Bei den realen Datensätzen besitzen dieselben Verfahren die besten Ränge, wobei SNE durch GPLVM ersetzt wird; die Reihenfolge ist: SM und EE und nach etwas Abstand mMDS, GPLVM, t-SNE und UKR. Allerdings kann SM bei zwei realen Datensätzen keine Einbettung berechnen. Da gute Ergebnisse bei realen Datensätzen für die praktische Anwendung eines Verfahrens wichtiger als ein guter Rang bei künstlichen Datensätzen sind, können zusammen mit den Ergebnissen der durchschnittlichen Ränge pro Metrik (Abschnitt 3.3.1) EE, t-SNE, UKR, mMDS und GPLVM als die besten nichtkonvexen Verfahren angesehen werden.

Es fällt auf, dass alle nichtkonvexen Verfahren bis auf SNE bei den realen Datensätzen einen besseren durchschnittlichen Rang erzielen (bei den konvexen Verfahren ist es nur die Hälfte) und die Abweichungen zwischen den Rangmittelwerten realer und künstlicher Datensätze insbesondere bei den MDS-Varianten und den auf einem LVM basierenden Verfahren groß sind. Der vermutliche Grund für diese großen Abweichungen ist, dass bei den realen Datensätzen bis zu sieben konvexe und nichtkonvexe Verfahren keine Einbettungen berechnen konnten, somit bei der Rangbildung nicht berücksichtigt werden und folglich die Ränge der schlechtesten übrigen Verfahren automatisch kleiner werden. Deshalb liegt die Vermutung nahe, dass die starke Abweichung der Mittelwerte bei den nichtkonvexen Verfahren, die bei den künstlichen Datensätzen sehr schlechte Ränge erreichen (z. B. Autoencoder, nMDS, FA, PPCA), zum Teil durch die fehlenden Ergebnisse verursacht werden. Die bisher besten nichtkonvexen Verfahren unterliegen diesem Effekt

vermutlich weniger stark, da sie tatsächlich bessere Ränge haben, von der automatischen Rangverschiebung aber vor allem schlechtere Verfahren profitieren.

Die Tabellenspalten für die einzelnen Datensätze zeigen, dass sich Verfahren mit ähnlichem Entwurf ähnlich verhalten. So haben mMDS und SM oft annähernd gleiche Ränge und sind beide bei Helix und MNIST schlechter als bei den übrigen Datensätzen. Bei UNN und itUKR ist Gleiches zu beobachten: Sie besitzen beide bei den Datensätzen Broken Swiss-roll, Twin-peaks, HD, Iris, Digits, HIVA und ORL einen schlechteren Rang. Außerdem zeigen die Ergebnisse, dass, wie bei den konvexen Verfahren, Fehler ausschließlich bei der Einbettung realer Datensätze entstehen, jedoch bei weniger Verfahren. Die Ursachen sind komplexe oder NaN-Werte in den Einbettungen von SM und FA, für die keine Gründe in den Fehlermeldungen angegeben wurden, und nicht lösbare Eigenwertprobleme bei MC.

Auf der Grundlage der Untersuchungen aus diesem Abschnitt und dem Abschnitt 3.3.1 werden EE, t-SNE, UKR, mMDS und GPLVM als beste nichtkonvexe Verfahren eingestuft. EE ist das zu bevorzugende Verfahren, da es weder bei bestimmten Datensätzen noch bezüglich irgendeiner Metrik einen schlechten durchschnittlichen Rang besitzt. Insgesamt erzielen nMDS und der Autoencoder für die vorliegenden Datensätze und Metriken keine erfolgreichen Ergebnisse, die LVM- und MM-basierten Verfahren außer GPLVM zeigen ebenfalls keine zufriedenstellenden Resultate. Die Ergebnisse der auf stochastischen Einbettungen beruhenden Verfarhen (UNN und itUKR) sind von mittlerer Qualität.

3.3.3 Vergleich der Visualisierungen

Mit Hilfe der visualisierten Einbettungen wird untersucht, ob unter den nichtkonvexen Verfahren für Visualisierungszwecke besonders geeignete sind. Die Einbettungen wurden wie in Abschnitt 3.2.3 beschrieben ausgewählt und wer-

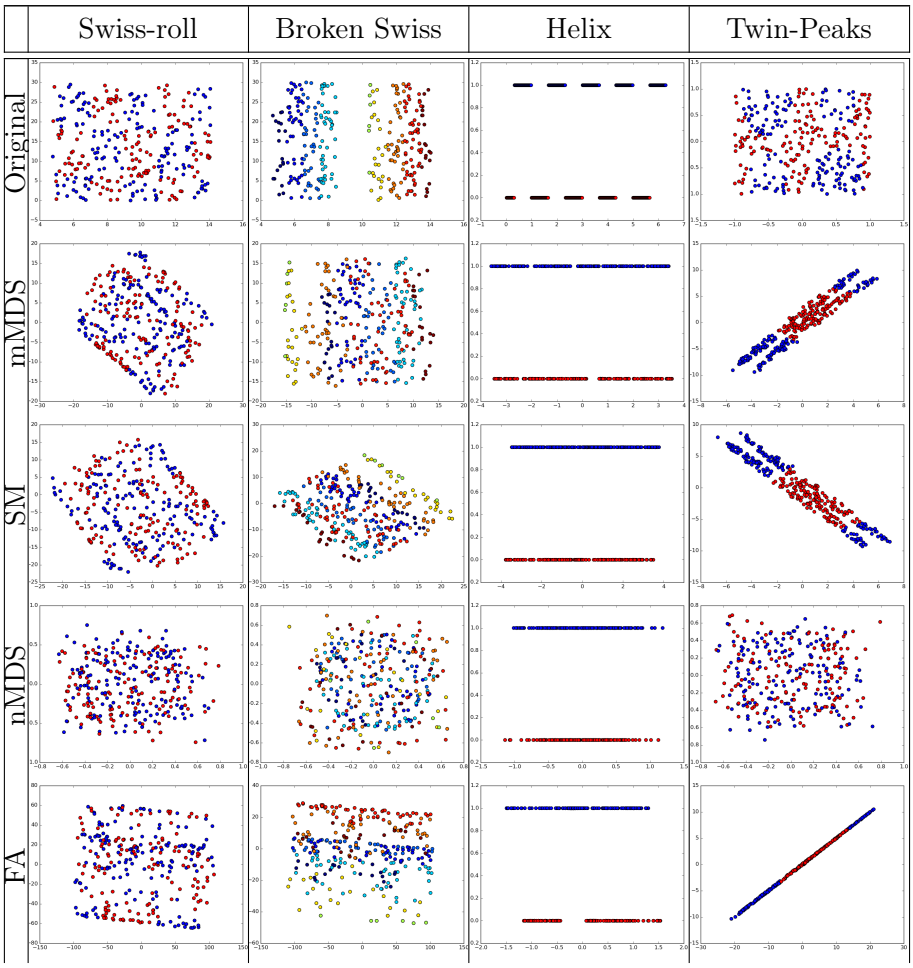
den ebenfalls danach untersucht, wie gut die Struktur der originalen Mannigfaltigkeit erhalten wird und Punkte unterschiedlicher Klassen getrennt werden.

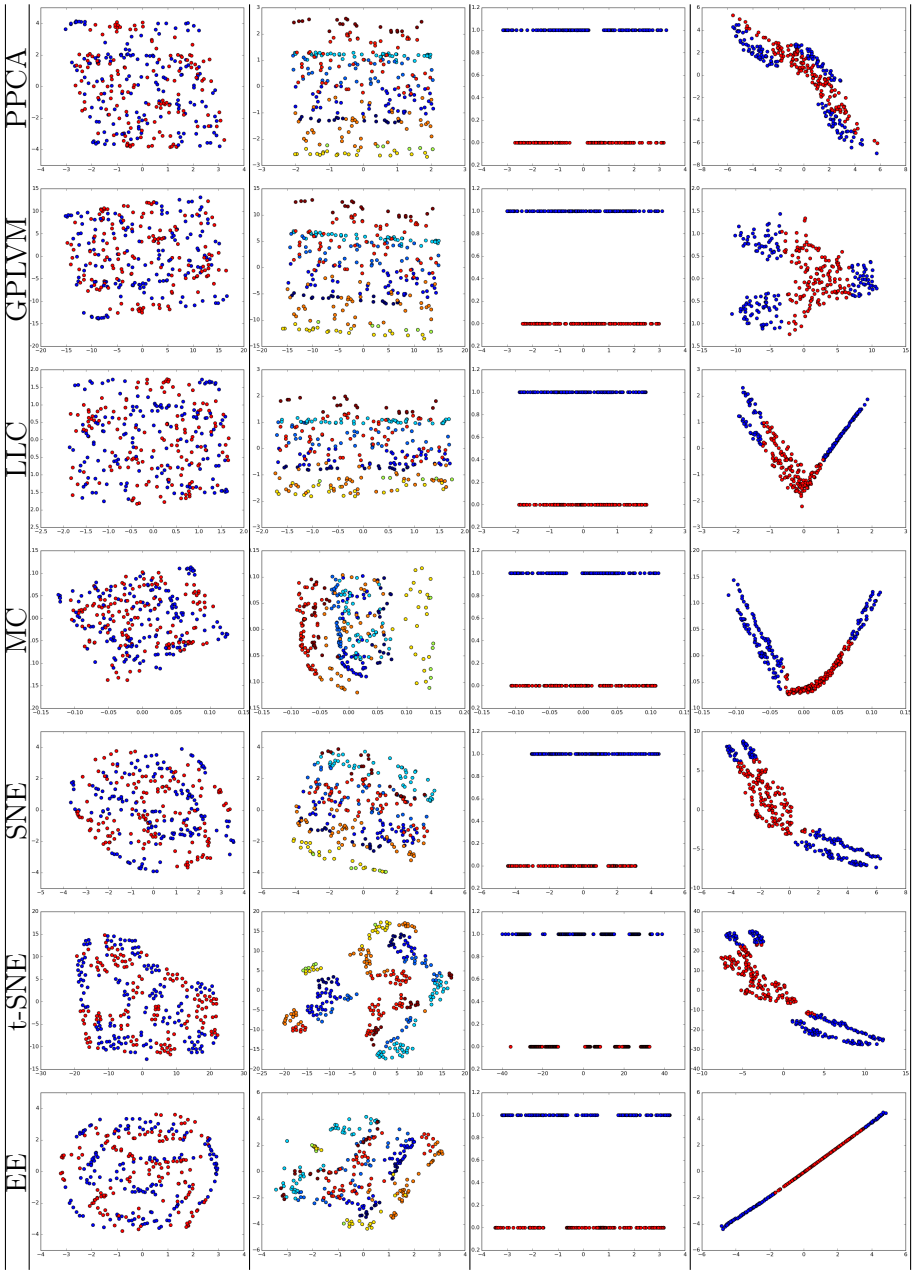
Künstliche Datensätze

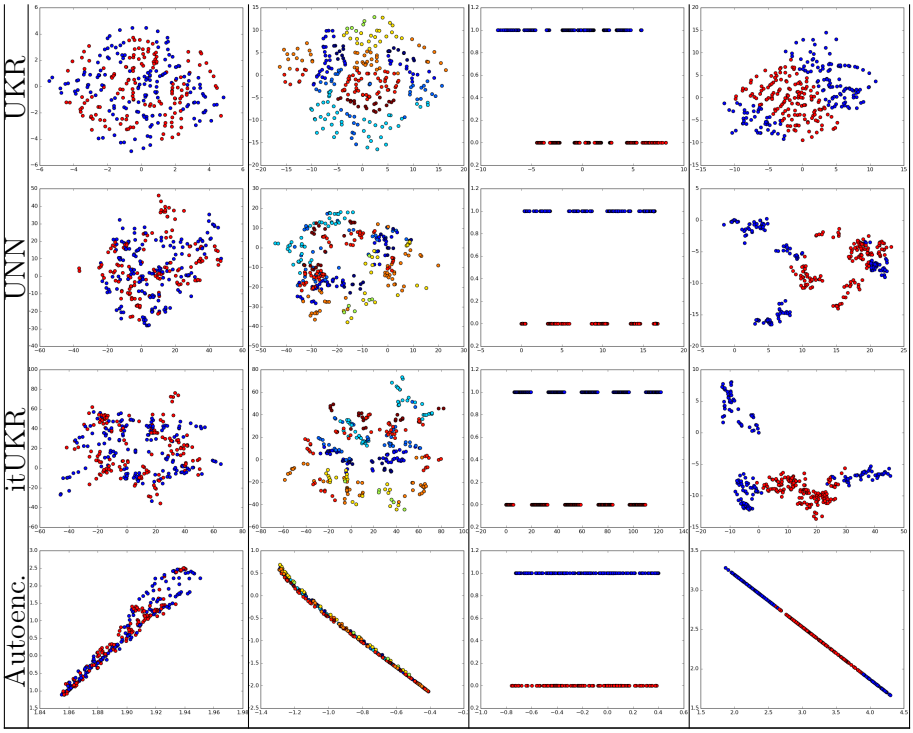
Bei Swiss-roll und Broken Swiss-roll zeigen die Verfahren ein ähnliches Verhalten: Anstatt die Rolle auseinander zu wickeln, wird sie flachgedrückt, sodass in der Mitte der sich ergebenden Mannigfaltigkeiten mehrere Schichten übereinander liegen. Die beste Trennung latenter Punkte unterschiedlicher Klassen wird bei Swiss-roll von UKR, t-SNE und EE erreicht, wobei die beiden letztgenannten große Löcher in der Mannigfaltigkeit aufweisen. Bei Broken Swiss-roll zerstören t-SNE und EE die Struktur sogar so weit, dass eher Häufungen von Punkten gleicher Klasse als ein Rechteck zu erkennen sind. Die Einbettungen von UNN und itUKR besitzen bei Swiss-roll und Broken Swiss-roll eine ähnlich fehlerhafte Struktur. itUKR erzeugt bei Helix die beste Einbettung, sie entspricht der originalen. Auch die Einbettung von UNN ist sehr gut, jedoch fehlt eine Gruppe blauer Punkte. Bei t-SNE und UKR ist ebenfalls die wechselweise Abfolge der Punkte unterschiedlicher Farbe zu sehen, diese sind aber nicht so klar getrennt wie bei itUKR. Allerdings spiegeln die durchschnittlichen Ränge diese Ergebnisse nicht wieder (Abbildung 3.7): itUKR besitzt bei Helix nur den zweitbesten durchschnittlichen Rang und UNN nur den fünftbesten, während EE den besten Rang hat, obwohl in dessen Einbettung keine klaren Farbwechsel zu sehen sind. Der Grund hierfür ist, dass itUKR bei Helix zwar bei jeder Metrik den ersten Rang erreicht, beim EKS jedoch nur den 24. Rang,[12] während EE den ersten belegt. Dies zeigt, dass nicht Metriken allein eine korrekte Einschätzung der Einbettungsqualität liefern. Bei Twin-peaks erzeugt kein Verfahren eine Einbettung, die der

[12] Diese Werte können der Datei all_optmeans_Masse-angepasst.ods der beigelegten DVD entnommen werden.

originalen ähnelt, jedoch sind Punkte unterschiedlicher Farbe nicht gemischt; einzig bei nMDS sind die Punkte durcheinander angeordnet. Die fehlende Trennung von latenten Punkten unterschiedlicher Farbe ist bei jeder Einbettung von nMDS zu beobachten. Außerdem fällt auf, dass itUKR, UNN, t-SNE und EE dazu neigen, Häufungen von Punkten mit großen Zwischenräumen in der Mannigfaltigkeit zu erzeugen.




PPCA
GPLVM
LLC
MC
SNE
t-SNE
EE



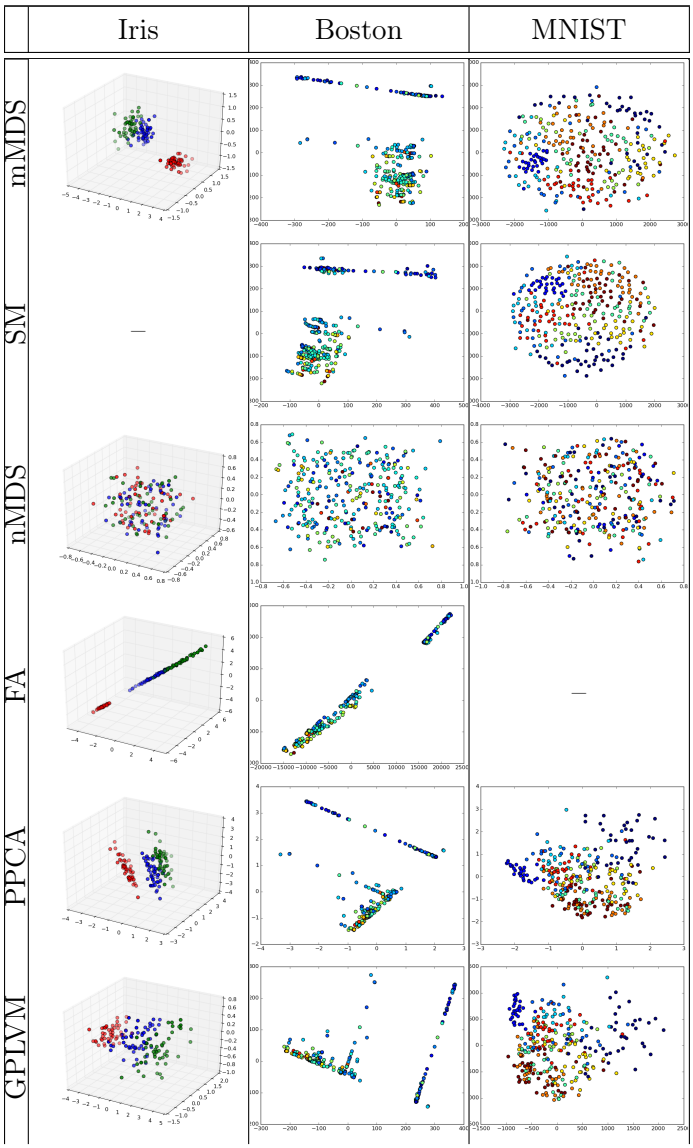
Tab. 3.4: Visualisierte Einbettungen nichtkonvexer Verfahren (künstliche Datensätze)

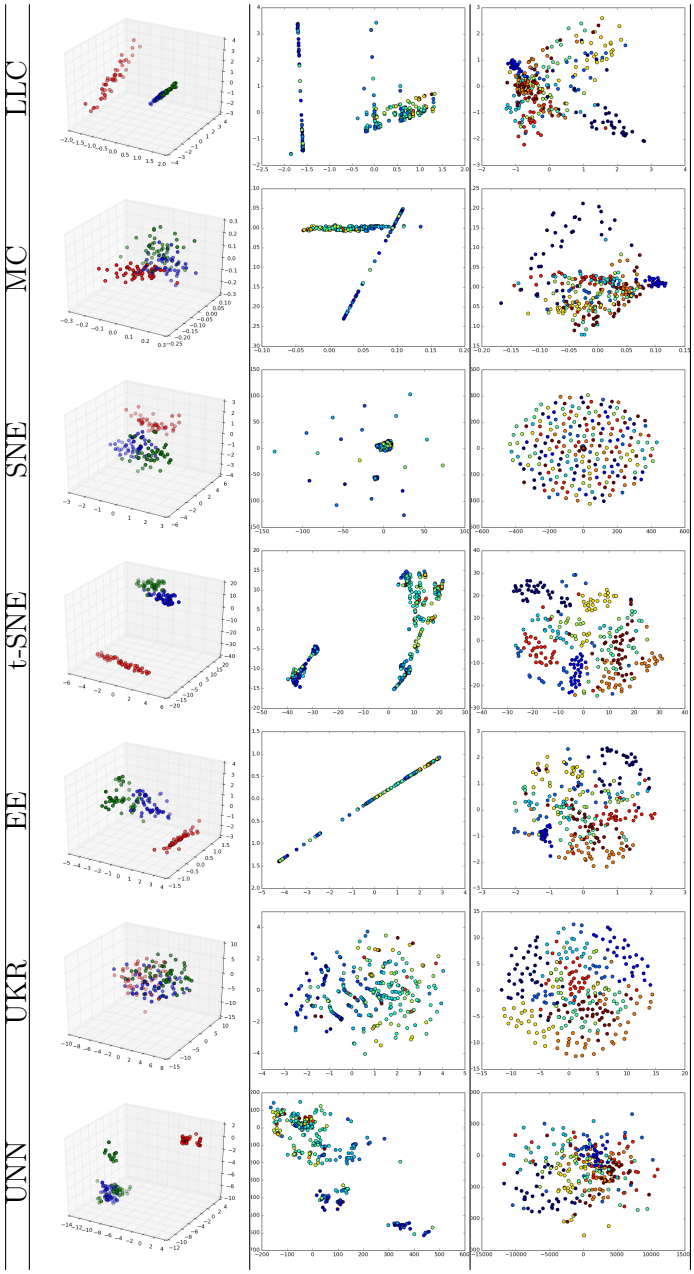
Reale Datensätze

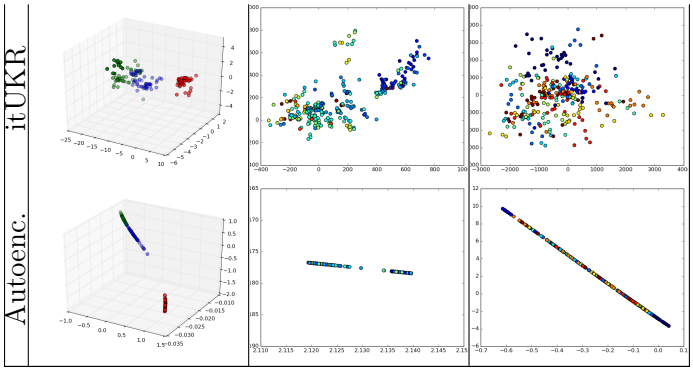
Da Iris ein Datensatz mit drei Klassen ist, wird eine Einbettung, die eine räumliche Trennung von Punkten unterschiedlicher Klassen zeigt, als gut angesehen (siehe Abschnitt 3.2.3). Räumlich getrennte Klassen werden am besten von PPCA, itUKR und t-SNE erreicht. Beim Autoencoder und FA fällt auf, dass sie zwar die Punkte unterschiedlicher Klassen nicht durchmischen, deren Einbettungen aber die Form einer Gerade im latenten Raum haben.

Bei Boston ist bei vielen Einbettungen, wie auch bei den konvexen Verfahren, eine Gruppe eher dunklerer Punkte und eine Gruppe hellerer Punkte zu erkennen. Auffällig ist, dass SNE die latenten Punkte häuft, was zur Interpretation der Daten wenig hilfreich ist. Auch bei der Einbettung von MNIST durch SNE wird ein Teil der Punkte in der Mitte der Einbettung auf derselben Stelle angeordnet. Bei MNIST ist die Einbettung von t-SNE am besten: Punkte gleicher Farbe sind nicht nur benachbart angeordnet, sondern meist auch räumlich voneinander getrennt. Dies ist bei keinem anderen nichtkonvexen oder konvexen Verfahren der Fall. EE weist ebenfalls eine räumliche Trennung der Klassen auf, die insgesamt aber weniger gut als die von t-SNE ist, da beispielsweise die gelben Punkte nicht beeinander angeordnet sind. Die Einbettung von UKR ist ebenfalls gut, da Punkte unterschiedlicher Farbe, obwohl räumlich nicht getrennt, wenig vermischt sind; mMDS und SM zeigen Ähnliches.

Insgesamt sind die visualisierten Einbettungen der nichtkonvexen Verfahren bei realen Datensätzen besser als bei künstlichen, da es bei Iris und MNIST jeweils mehrere Verfahren mit guter Trennung der Klassen gibt, während bei den künstlichen Datensätzen nur Helix von nur einem Verfahren zufriedenstellend visualisiert wird. Diese Beobachtung stimmt mit dem Ergebnis aus Abschnitt 3.3.2 überein, dass außer SNE alle nichtkonvexen Verfahren auf realen Datensätzen einen besseren durchschnittlichen Rang als auf künstlichen besitzen. Aufgrund guter Einbettungen bei MNIST und Iris kann t-SNE als geeignetestes Visualisierungsverfahren eingestuft werden. Die schlechtesten Visualisierungen bei den nichtkonvexen Verfahren erzeugen nMDS und der Autoencoder: nMDS durchmischt willkürlich Punkte verschiedener Klassen, der Autoencoder erzeugt linienartige Einbettungen statt erkennbarer Strukturen.


Iris
Boston
MNIST
mMDS
SM
nMDS
FA
PPCA
GPLVM


LLC
MC
SNE
t-SNE
EE
UKR
UNN



Tab. 3.5: Visualisierte Einbettungen nichtkonvexer Verfahren (reale Datensätze)

3.3.4 Zusammenfassung

Die Ergebnisse der Abschnitte zum Vergleich nichtkonvexer Verfahren zeigen, dass EE, t-SNE, UKR, mMDS und GPLVM unter Berücksichtigung ihrer durchschnittlichen Ränge bezüglich der Metriken und Datensätze die besten sind. Wird ein Dimensionsreduktionsverfahren für die Visualisierung eines hoch-dimensionalen Datensatzes benötigt, bietet sich von den nichtkonvexen Verfaren t-SNE an. Da in dieser Arbeit für jeden Datensatz jedoch pro Verfahren nur die Einbettung für eine einzige Parameterbelegung visualisiert wurde, besteht die Möglichkeit, dass die Eignung eines anderen Verfahrens unerkannt geblieben ist.

3.4 Gemeinsamer Vergleich

In diesem abschließenden Vergleich werden die durchschnittlichen Ränge der besten konvexen und nichtkonvexen Verfahren ausgewertet (Abschnitt 3.4.1).

Verfahren	ENX	EKS	DSRE	E1NN	CRR	EKNN	Ø
PCA	10.8	12.0	7.0	15.4	16.9	11.5	12.3
Isomap	12.8	13.7	9.2	8.3	9.7	15.0	11.4
MLLE	10.8	10.3	15.9	7.6	10.9	10.0	10.9
LE	10.8	13.4	15.2	7.5	6.5	16.0	11.6
mMDS	7.4	8.2	2.7	14.1	11.5	23.0	11.1
GPLVM	10.8	11.4	6.6	14.7	15.3	10.0	11.5
t-SNE	4.9	8.2	17.2	6.1	5.4	5.0	7.8
EE	5.1	7.1	8.5	4.0	4.2	11.0	6.6
UKR	12.2	3.0	15.9	6.5	9.0	1.5	8.0

Abb. 3.8: Durchschnittliche Ränge der besten konvexen und nichtkonvexen Verfahren pro Metrik; kleine Werte sind gut (gelb hinterlegt)

Anschließend werden die Qualitätsunterschiede aller Verfahren und die statistische Signifikanz der Metrikwerte untersucht (Abschnitte 3.4.2 und 3.4.3). Der Abschnitt schließt mit einer Zusammenfassung der Ergebnisse des experimentellen Vergleichs (Abschnitt 3.4.4).

3.4.1 Vergleich hinsichtlich der Eigenschaften und Datensätze

Die Untersuchungen in den vorigen Abschnitten ergaben, dass hinsichtlich der durchschnittlichen Ränge MLLE, Isomap, PCA und LE die besten konvexen und EE, t-SNE, UKR, mMDS und GPLVM die besten nichtkonvexen Dimensionsreduktionsverfahren sind. Die Zusammenstellung der durchschnittlichen Ränge dieser Verfahren pro Metrik aus den Abbildungen 3.2 und 3.5 in einer gemeinsamen Abbildung (3.8) zeigt, dass EE, t-SNE und UKR gegenüber den anderen Verfahren deutlich bessere Ränge aufweisen. Die durchschnittlichen Ränge pro Datensatz (Abbildung 3.9) zeigen zum einen, dass bis auf Isomap und MLLE alle diese Verfahren auf realen Datensätzen bessere Ränge erreichen, und zum anderen, dass die nichtkonvexen Verfahren auf den realen Datensätzen deutlich besser als die konvexen abschneiden. Lediglich PCA erreicht in etwa den gleichen Rang wie die nichtkonvexen Verfahren. Somit kann bezüglich der durchschnittlichen Ränge EE als das mit Abstand beste in dieser Arbeit untersuchte Dimensionsreduktionsverfahren angesehen werden,

Verfahren	Swiss	Broken	Helix	Twin	HD	Iris	Boston	Digits	RCT	MNIST	HIVA	CNS	ORL	Ø(k.)	Ø(r.)	Abw.
PCA	21.4	20.6	23.0	11.2	12.4	5.6	15.3	4.2	3.8	18.6	7.8	6.2	7.0	17.7	8.6	9.2
Isomap	8.6	4.8	9.6	6.6	18.4	12.0	13.3	14.8	8.8	14.0	8.0	7.0	17.0	9.6	11.9	2.3
MLLE	10.6	6.4	8.4	12.4	11.0	17.2	10.3	7.6	17.3	10.2	7.2	14.8	13.2	9.8	12.2	2.5
LE	8.2	11.4	8.2	12.8	17.2	16.2	21.0	8.4	9.0	5.2	6.6	10.4	11.0	11.6	11.0	0.6
mMDS	9.0	11.6	20.2	5.4	8.6	8.0	11.3	5.2	4.3	11.2	5.2	6.4	6.0	11.0	7.2	3.8
GPLVM	21.0	20.2	23.0	11.0	11.0	5.6	14.5	3.8	3.3	17.4	4.6	6.2	7.0	17.2	7.8	9.4
t-SNE	6.2	5.4	7.0	9.4	18.8	11.2	6.8	9.6	7.3	1.8	11.8	6.8	7.0	9.4	7.8	1.6
EE	6.0	4.6	4.0	10.8	9.2	6.8	8.5	5.0	11.3	2.8	4.2	5.2	1.8	6.9	5.7	1.2
UKR	5.8	7.0	7.8	13.4	21.4	16.4	8.8	7.6	8.3	2.8	6.6	7.2	6.2	11.1	8.0	3.1

Abb. 3.9: Durchschnittliche Ränge der besten konvexen und nichtkonvexen Verfahren pro Datensatz; kleine Werte sind gut (gelb hinterlegt)

mMDS, GPLVM, t-SNE, UKR und PCA sind ebenfalls sehr gut.

3.4.2 Vergleich des Anteils am minimalen Fehler (AMF)

In den bisherigen Untersuchungen wurde lediglich die Rangfolge der Verfahren betrachtet. In diesem Abschnitt wird untersucht, wie stark sich die Qualität der Einbettungen unterscheidet. Dafür wird, ähnlich wie in [FD+14, S. 3163], für jedes Verfahren V pro Metrik M der durchschnittliche Anteil am besten, d. h. minimalen, Metrikwert berechnet. Dieser wird, da die Metriken Fehlermaße darstellen, als *Anteil am minimalen Fehler* (AMF) bezeichnet:

$$\mathrm{AMF}_V(M) = \frac{1}{13} \sum_{D \in \{\text{Swiss-roll},\ldots,\text{ORL}\}} \frac{\text{von } V \text{ erzielter Wert für } M \text{ auf } D}{M^*}, \tag{3.14}$$

wobei M^* der beste Wert für M ist, den eines der Verfahren auf dem Datensatz D erreicht. Die AMF-Werte der Verfahren sind im linken Teil der Abbildung 3.10 zu sehen. Der AMF-Wert von 1.8 von SM für den ENX beispielsweise besagt, dass die ENX-Werte von SM im Mittel das 1.8fache des jeweils besten ENX-Wertes betragen. Es werden nicht die absoluten Werte der Metriken verglichen, da sie unterschiedliche Wertebereiche besitzen und somit unvergleichbar sind.

Verfahren	ENX	Verfahren	EKS	Verfahren	DSRE	Verfahren	E1NN	Verfahren	CRR	Verfahren	EKNN
SM	**1.8**	SM	**1.1**	UKR	**1.2**	t-SNE	**1.3**	Isomap	**1.6**	MVU	**1.0**
mMDS	1.9	mMDS	1.3	t-SNE	1.4	UKR	2.1	UKR	1.6	UKR	1.0
PCA	2.0	CE	1.5	EE	1.7	EE	2.7	EE	1.6	SM	1.1
GPLVM	2.0	MVU	1.6	itUKR	2.1	LE	3.0	t-SNE	1.7	NPE	1.1
MVU	2.1	GPLVM	3.2	UNN	2.3	itUKR	3.3	LE	2.0	GPLVM	1.1
CE	2.1	PCA	3.2	LE	2.6	Isomap	3.8	itUKR	2.0	PCA	1.1
t-SNE	2.4	LLTSA	15.7	MLLE	2.7	MLLE	3.9	MLLE	2.1	HLLE	1.1
EE	2.5	FA	76.8	SNE	2.7	UNN	3.9	DM	2.1	SNE	1.1
HLLE	2.8	Isomap	77.3	mMDS	2.9	LTSA	4.0	HLLE	2.1	CE	1.1
LLTSA	3.0	EE	132.9	Isomap	2.9	LLC	4.2	LTSA	2.2	LPP	1.1
SNE	3.1	HLLE	224.0	LLE	3.1	LLE	4.3	LLE	2.3	LLTSA	1.1
LE	3.6	NPE	263.9	GPLVM	3.1	mMDS	4.3	SNE	2.3	t-SNE	1.1
Isomap	3.8	LPP	287.1	PCA	3.1	DM	4.4	UNN	2.4	mMDS	1.1
DM	4.2	t-SNE	306.1	LTSA	3.2	HLLE	4.9	LLC	2.5	Isomap	1.2
LPP	4.4	Autoenc.	1161.5	DM	3.3	GPLVM	5.0	GPLVM	2.6	LE	1.2
FA	4.4	itUKR	1272.2	SM	3.4	PCA	5.0	PCA	2.6	itUKR	1.2
NPE	4.4	DM	1798.9	LLC	3.5	SM	5.1	CE	2.6	EE	1.2
UKR	4.5	LTSA	1860.3	HLLE	3.7	CE	5.2	mMDS	2.7	FA	1.2
MLLE	4.6	UKR	2088.0	MVU	3.8	MVU	5.4	MVU	2.7	KPCA	1.3
LLE	4.8	LE	2128.9	CE	3.9	LPP	5.5	SM	2.8	UNN	1.3
UNN	4.8	LLC	2357.4	LLTSA	4.2	PPCA	5.6	LLTSA	2.9	Autoenc.	1.3
itUKR	4.8	UNN	2389.9	FA	4.6	MC	5.6	FA	2.9	MLLE	1.3
LTSA	5.1	MC	2434.6	MC	4.6	FA	5.6	LPP	3.1	LLE	1.3
PPCA	5.3	LLE	2531.1	Autoenc.	4.7	NPE	6.2	KPCA	3.2	DM	1.7
KPCA	5.5	MLLE	2658.8	LPP	5.0	SNE	7.2	MC	3.3	LTSA	3.0
MC	5.6	PPCA	2984.3	NPE	5.2	KPCA	9.3	Autoenc.	3.4	LLC	4.6
LLC	5.6	SNE	3478.2	PPCA	5.6	LLTSA	9.4	NPE	3.5	MC	6.2
Autoenc.	6.0	KPCA	3649.6	KPCA	6.1	Autoenc.	11.0	PPCA	3.9	PPCA	12.9
nMDS	10.31	nMDS	6586.4	nMDS	31.5	nMDS	29.61	nMDS	13.8	nMDS	18.1

Verfahren	Ø
mMDS	**2.4**
SM	2.5
CE	2.7
MVU	2.8
GPLVM	2.8
PCA	2.8
LLTSA	6.1
Isomap	15.1
FA	15.9
EE	23.8
HLLE	39.8
NPE	47.4
LPP	51.0
t-SNE	52.4
Autoenc.	198.0
itUKR	214.3
DM	302.4
LTSA	313.0
UKR	349.7
LE	356.9
LLC	396.3
UNN	400.8
MC	410.0
LLE	424.5
MLLE	445.6
PPCA	502.9
SNE	582.4
KPCA	612.5
nMDS	1115.0

Verfahren	Ø o. EKS
t-SNE	1.6
EE	1.9
UKR	2.1
LE	2.5
mMDS	2.6
Isomap	2.6
itUKR	2.7
GPLVM	2.7
PCA	2.8
SM	2.8
MLLE	2.9
UNN	2.9
HLLE	2.9
CE	3.0
MVU	3.0
DM	3.1
LLE	3.2
SNE	3.3
LTSA	3.5
FA	3.7
LPP	3.8
NPE	4.1
LLC	4.1
LLTSA	4.1
KPCA	5.0
MC	5.1
Autoenc.	5.3
PPCA	6.7
nMDS	20.7

Abb. 3.10: Anteil am minimalen Fehler (AMF) (links), Durchschnitt aller AMF-Werte eines Verfahrens (Mitte), Durchschnitt ohne EKS-Wert (rechts); kleine Werte sind gut, konvexe Verfahren blau hinterlegt

Die gemittelten AMF-Werte der Verfahren zeigen ein überraschendes Ergebnis (mittlerer Teil der Abbildung 3.10): Die Verfahren EE, t-SNE und UKR, die hinsichtlich der durchschnittlichen Ränge zu den besten Verfahren gehören, sind nicht unter den Verfahren mit dem besten, d. h. geringsten durchschnittlichen AMF, sondern die besten sind mMDS, GPLVM und PCA. Aufgrund ihrer in Abschnitt 3.2.2 beschriebenen Schwierigkeiten bei realen Datensätzen, werden SM, CE und MVU trotz ihres guten AMF-Durchschnitts nicht zu den besten Verfahren gezählt. Der Grund für das schlechtere Abschneiden von EE, t-SNE und UKR ist ihr im Vergleich zu den anderen Metriken großer AMF beim EKS. Zwar wurde bereits in Abschnitt 3.3.1 bei der Untersuchung der Ranghäufigkeiten festgestellt, dass die Einbettungen von t-SNE und UKR bezüglich des EKS nicht zu den besten Verfahren gehören, allerdings ist bemerkenswert, dass EE mit einem durchschnittlichen Rang von 8.5 beim EKS zu den besseren Verfahren gehört und dennoch einen großen AMF beim EKS hat. Auch die AMF-Werte der meisten anderen Verfahren sind beim EKS deutlich größer als bei anderen Metriken, sodass der EKS einen starken Einfluss auf den durchschnittlichen AMF-Wert hat. Dadurch werden mMDS und SM bevorzugt, da der EKS ihrer Zielfunktion entspricht (siehe Abschnitt 3.1.1). Allerdings entspricht der Einfluss des EKS, den es aufgrund seines großen Wertebereiches auf den durchschnittlichen AMF hat, nicht dessen Wichtigkeit bei der Beurteilung der Qualität eines Dimensionsreduktionsverfahrens, da er ein globales Topologieerhaltungsmaß ist. Nach [MPH09, S. 18] wird jedoch die Bewahrung lokaler Strukturen, z. B. Nachbarschaften, als wichtiger erachtet, da diesen beispielsweise eine große Bedeutung zukommt, wenn das Dimensionsreduktionsverfahren Muster für eine Klassifikation oder Regression vorverarbeiten soll.

Folglich sollten bei der Auswahl eines Dimensionsreduktionsverfahrens nicht ausschließlich der durchschnittliche AMF, sondern auch die einzelnen AMF-Werte der Metriken, die für den Anwendungszweck besonders relevant sind, in Betracht gezogen werden. Im rechten Teil der Abbildung 3.10 werden bei-

spielhaft die Mittelwerte ohne die Werte für den EKS gezeigt, was zu einer gänzlich anderen Reihenfolge der Verfahren führt: t-SNE, EE und UKR nehmen die vordersten Plätze ein, allerdings sind die Unterschiede zu den übrigen Verfahren relativ gering.

3.4.3 Statistische Signifikanz der Metrikwerte

Die Fragestellung, ob sich die Ergebnisse verschiedener Algorithmen statistisch signifikant voneinander unterscheiden, tritt im Bereich des maschinellen Lernens häufig auf. [SIL15] gibt einen Überblick über verschiedene Testprobleme und geeignete statistische Tests. In dieser Arbeit wird der *Friedman-Test* verwendet, der sich laut [SIL15] für das vorliegende Testproblem, dem Vergleich verschiedener Algorithmen auf unterschiedlichen Datensätzen, eignet. Die Nullhypothese des Friedman-Tests besagt, dass alle Verfahren die gleiche Qualität besitzen. Beim Friedman-Test werden die Verfahren entsprechend ihres gemessenen Qualitätswertes getrennt für jeden Datensatz in eine Rangfolge gebracht und die erzielten Ränge pro Verfahren über alle Datensätze gemittelt. Über diese durchschnittlichen Ränge wird die Teststatistik berechnet [SIL15].

Da in der vorliegenden Arbeit sechs verschiedene Metriken als Qualitätsmaße verwendet werden, wird für jede Metrik ein Friedman-Test[13] durchgeführt, außer für den EKNN, da dieser nur bei zwei Datensätzen erhoben werden konnte (siehe Abschnitt 3.1.1) und somit nicht genügend Daten für einen aussagekräftigen Test vorliegen. Die Ergebnisse der Tests sind sehr geringe p-Werte (Tabelle 3.6), die deutlich unter üblicherweise verwendeten Signifikanzniveaus von 0.05 oder 0.01 liegen. Aus diesem Grund wird die Nullhypothese abgelehnt und davon ausgegangen, dass die in den vorigen Abschnitten dargelegten Unterschiede der Verfahren statistisch signifikant sind, die un-

[13] Die Friedman-Tests wurden unter Verwendung des Python-Moduls `stats` von SciPy implementiert (`http://scipy.org/`, letzter Zugriff am 2.12.2016).

tersuchten Dimensionsreduktionsverfahren also nicht zufällig unterschiedliche Metrikwerte erzielt haben.

	ENX	EKS	DSRE	E1NN	CRR
p-Wert	2.66e-28	1.65e-43	4.57e-66	1.94e-36	7.58e-36

Tab. 3.6: p-Werte der Friedman-Tests

3.4.4 Zusammenfassung

Der gemeinsame Vergleich konvexer und nichtkonvexer Verfahren zeigt, dass die Qualität der Einbettungen von Verfahren, die häufig gute durchschnittliche Ränge besitzen, schlechter als die von Verfahren sein kann, die weniger gute Ränge besitzen. Interessant ist, dass mit mMDS und PCA zwei der ältesten Dimensionsreduktionsverfahren unter den Verfahren mit dem besten mittleren AMF sind (mMDS, GPLVM, PCA). Zudem wurde gesehen, dass abhängig davon, welche Metriken bei bestimmten Anwendungszwecken die Eignung eines Verfahrens widerspiegeln, andere Verfahren eine bessere Wahl darstellen. Außerdem wurde die statistische Signifikanz der Unterschiede der Metrikwerte der Verfahren gezeigt.

4 Schluss

Auf der Grundlage der im experimentellen Vergleich gewonnenen Ergebnisse werden entsprechend der Zielsetzung dieser Arbeit, die Entscheidung für den Einsatz eines bestimmten Dimensionsreduktionsverfahrens mit Hilfe quantifizierter Eigenschaften der Verfahren zu vereinfachen, im nachfolgenden Abschnitt (4.1) Empfehlungen zur Auswahl von Dimensionsreduktionsverfahren gegeben. Zusätzlich werden in Abschnitt 4.2 Erweiterungsmöglichkeiten dieser Arbeit aufgezeigt und in Abschnitt 4.3 die wesentlichen Resultate dieser Arbeit zusammengefasst.

4.1 Empfehlungen zur Wahl von Dimensionsreduktionsverfahren

In dieser Arbeit wurde dargelegt, dass kein Dimensionsreduktionsverfahren existiert, das für jede Metrik und jeden Datensatz optimal geeignet ist. Dies war zu erwarten, da bereits die *no-free-lunch*-Theoreme [WM97] besagen, dass, wenn ein oder mehrere Optimierungsalgorithmen hinsichtlich einer Teilmenge aller möglicher Optimierungsprobleme besser als ein oder mehrere andere Algorithmen sind, diese anderen Verfahren auf allen übrigen Optimierungsproblemen besser abschneiden. Folglich ist die mit einer Metrik gemessene und über alle möglichen Optimierungsprobleme gemittelte Qualität aller Optimierungsalgorithmen gleich [WM97]. Da Dimensionsreduktionsverfahren ebenfalls Optimierungsprobleme lösen, folgt daraus, dass aus dem Verhalten eines Dimensionsreduktionsverfahrens bei bestimmten Datensätzen nicht auf

dessen Verhalten bei einem beliebigen anderen Datensatz geschlossen werden kann. Dies bedeutet, dass bei jeder geplanten Anwendung einer Dimensionsreduktion entschieden werden muss, welches Verfahren am sinnvollsten ist. Die in der Praxis am häufigsten vorkommenden Anwendungsfälle, in denen Dimensionsreduktionsverfahren benötigt werden, sind die beiden folgenden: der Einsatz von Dimensionsreduktionsverfahren als Referenzverfahren, um die Qualität eines neu entwickelten Dimensionsreduktionsverfahren zu evaluieren, und die Reduktion eines hoch-dimensionalen Datensatzes, um dessen praktische Nutzung zu ermöglichen (z. B. bei der Erforschung von Gesundheitsdaten). Für diese Anwendungsfälle werden im Folgenden Empfehlungen zur Auswahl von Dimensionsreduktionsverfahren gegeben.

Werden im ersten Anwendungsfall Datensätze eingesetzt, die auch in der vorliegenden Arbeit verwendet wurden, kann diese Arbeit eine direkte Hilfestellung bei der Auswahl von Referenzverfahren geben, da das Verhalten der Verfahren auf diesen Datensätzen bekannt ist.[1] Als Referenzverfahren sollten die Verfahren ausgewählt werden, die entsprechend der Abbildungen 3.4, 3.7 und 3.9 auf diesen Datensätzen die besten durchschnittlichen Ränge erzielen. Die AMF-Werte können zwar als nachrangige Kriterien ebenfalls berücksichtigt werden, sind in diesem Fall jedoch weniger hilfreich, da sie durch alle in dieser Arbeit verwendeten Datensätze und nicht nur durch die für den Anwendungsfall ausgewählten beeinflusst werden.

Im zweiten Anwendungsfall, in welchem die Dimensionsreduktion für einen Datensatz ausgeführt werden soll, der in dieser Arbeit nicht berücksichtigt wurde, kann entsprechend der no-free-lunch-Theoreme auf Grundlage der Ergebnisse dieser Arbeit keine exakte Angabe gemacht werden, welches der Verfahren das beste sein wird. Daher wäre es ideal, wenn alle in dieser Arbeit betrachteten Dimensionsreduktionsverfahren auf dem neuen Datensatz ausgeführt würden und das beste ausgewählt würde. In der Praxis steht jedoch

[1] Genau genommen nur auf der zufällig ausgewählten Teilmenge von 300 Mustern.

selten ausreichend Zeit für einen derartig groß angelegten Vergleich zur Verfügung, sodass nur eine Teilmenge der Verfahren in diese Evaluation einbezogen werden kann. Die Festlegung dieser Teilmenge anhand der in dieser Arbeit ermittelten und in Abbildung 3.10 aufgeführten AMF-Werte erscheint als sinnvolles Vorgehen, da diese das Verhalten der Verfahren über alle Datensätze mitteln und zu erwarten ist, dass sich die untersuchten Verfahren zumindest auf ähnlichen Datensätzen ähnlich verhalten werden. Die durchschnittlichen Ränge pro Metrik (Abbildungen 3.2, 3.5 und 3.8) sind hier weniger aussagekräftig, da vermutlich in den meisten Anwendungsfällen ein Verfahren, das im Mittel einen geringen Qualitätsunterschied zum besten Verfahren aufweist, einem Verfahren vorgezogen wird, das häufig das beste Verfahren ist, aber einen höheren Qualitätsunterschied besitzt. Für die auf den AMF-Werten basierende Auswahl der Verfahren sollte festgelegt werden, ob alle betrachteten Metriken oder beispielsweise nur die Metriken, welche die Informationserhaltung messen, für den speziellen Anwendungszweck geeignete Qualitätskriterien darstellen. Werden die AMF-Werte der Verfahren über die gewählten relevanten Metriken gemittelt, sind die Qualitätsunterschiede der Verfahren gut zu erkennen. Zusätzlich sollten die durchschnittlichen Ränge der Verfahren auf den unterschiedlichen Datensätzen berücksichtigt werden, um insbesondere die Eignung der Verfahren für reale Datensätze zu prüfen.

Für die Auswahl eines Dimensionsreduktionsverfahrens zur Visualisierung eines Datensatzes kann aufgrund der insgesamt wenig überzeugenden Visualisierungsergebnisse keine allgemeine Empfehlung gegeben werden. Lediglich t-SNE scheint aufgrund der guten räumlichen Trennung von Punkten unterschiedlicher Klassen am ehesten zur Visualisierung geeignet zu sein.

4.2 Erweiterungsmöglichkeiten des Vergleichs

Um die Eignung der Verfahren speziell für die Visualisierung von Datensätzen zu untersuchen, kann das Evaluationskonzept dieser Arbeit um geringfügige

Anpassungen erweitert werden. Zum einen sollten die Datensätze nicht auf ihre geschätzte intrinsische, sondern auf eine visualisierbare Dimensionalität reduziert werden; zwei Dimensionen sind oft die bevorzugte Wahl. Zum anderen sollte bei jedem Verfahren pro Datensatz nicht nur wie in dieser Arbeit die Einbettung der Parameterbelegung betrachtet werden, bei der das Verfahren den besten EKS-Wert erreicht, sondern zusätzlich die Einbettungen bei den bezüglich der anderen Metriken besten Parameterbelegungen. Außerdem wäre eine größere Anzahl an Mustern sinnvoll, um in den Visualisierungen Gruppen von Mustern gleicher Farbe eindeutiger erkennen zu können.

Daneben könnte der experimentelle Vergleich dieser Arbeit künftig um hybride Dimensionsreduktionsverfahren oder submanifold learner erweitert werden, die ein eher jüngeres Forschungsgebiet der Dimensionsreduktion darstellen, aber, wie beispielsweise in [Kra+16; Kra13] dargelegt wird, vielversprechende Ergebnisse liefern.

4.3 Zusammenfassung

In dieser Arbeit wurden die Einbettungen von 29 Dimensionsreduktionsverfahren für 13 reale und künstliche Datensätze anhand von sechs Metriken bewertet und zusätzlich visuell miteinander verglichen. Mit der großen Anzahl an verwendeten Verfahren, Metriken und Datensätzen geht der Umfang dieser Arbeit über bisherige experimentelle Vergleiche von Dimensionsreduktionsverfahren hinaus und legt gegenüber einigen anderen Arbeiten den Schwerpunkt nicht auf den visuellen, sondern den quantitativen Vergleich der Verfahren. Die in Tabellen zusammengestellten quantifizierten Eigenschaften der Verfahren ermöglichen dem Anwender eine gezieltere Auswahl eines Dimensionsreduktionsverfahrens als es bisher möglich war. Die Frage des Anwenders nach dem besten Verfahren für seinen Anwendungszweck kann in dem Fall treffend beantwortet werden, wenn die Dimensionsreduktion für einen in dieser Arbeit verwendeten Datensatz ausgeführt werden soll. Wenn ein in dieser

Arbeit nicht genutzter Datensatz reduziert werden soll, lässt sich wegen der no-free-lunch-Theoreme kein konkretes Verfahren angeben. Stattdessen kann auf Grundlage der Ergebnisse dieser Arbeit eine Vorauswahl an Verfahren getroffen werden, unter denen der Anwender in praktischen Experimenten das beste für den zu reduzierenden Datensatz ermitteln sollte. Somit gibt diese Arbeit sowohl Forschern im Bereich der Dimensionsreduktion als auch Anwendern in der Praxis eine Entscheidungshilfe zur Wahl geeigneter Dimensionsreduktionsverfahren an die Hand.

Erweiterungmöglichkeiten dieser Arbeit bestehen hinsichtlich des Vergleichs der Eignung der Verfahren zur Visualisierung von Datensätzen. Zudem wird mit der fortschreitenden Entwicklung weiterer Dimensionsreduktionsverfahren, insbesondere der hybriden Verfahren und der submanifold learner, ein experimenteller Vergleich auch dieser Verfahren notwendig werden.

A Wertebereiche der Parameter

Verfahren	Wertebereiche der Parameter
PCA	–
Isomap	$k \in \{5, 7, 9, 11, 13, 15, 50, 100\}$
KPCA	$i \in \{100, 200, 300, 400, 500\}$
MVU	$k \in \{5, 7, 9, 11, 13, 15, 50, 100\}$
DM	$t \in \{1, 10, 30, 50, 70, 90\}, \sigma \in \{0.2, 1.0, 5.0\}$
LLE	$k \in \{5, 7, 9, 11, 13, 15, 50, 100\}$
MLLE	$k \in \{5, 7, 9, 11, 13, 15, 50, 100\}$
NPE	$k \in \{5, 7, 9, 11, 13, 15, 50, 100\}$
LE	$k \in \{5, 9, 13, 50, 100\}, \sigma \in \{0.2, 1.0, 5.0\}$
LPP	$k \in \{5, 9, 13, 50, 100\}, \sigma \in \{0.2, 1.0, 5.0\}$
CE	$k \in \{5, 7, 9, 11, 13, 15, 50, 100\}$
HLLE	$k \in \{5, 7, 9, 11, 13, 15, 50, 100\}$
LTSA	$k \in \{5, 7, 9, 11, 13, 15, 50, 100\}$
LLTSA	$k \in \{5, 7, 9, 11, 13, 15, 50, 100\}$

Tab. A.1: Wertebereiche der Parameter der konvexen Verfahren

Verfahren	Wertebereiche der Parameter
mMDS	–
SM	–
nMDS	–
FA	–
PPCA	$i \in \{100, 200, 300, 400, 500\}$
GPLVM	$\sigma \in \{0.2, 0.5, 1.0, 2.5, 5.0\}$
LLC	$k \in \{5, 13, 100\}, a \in \{2, 9, 20\}, i \in \{200, 400\}$
MC	$a \in \{2, 7, 12, 20\}, i \in \{200, 400\}$
SNE	$p \in \{5, 10, 15, 20, 25, 30, 35, 40, 45, 50\}$
t-SNE	$p \in \{5, 10, 15, 20, 25, 30, 35, 40, 45, 50\}$
EE	$h \in \{0.01, 0.1, 0.0, 10.0, 100.0\}, p \in \{5, 25, 50\}$
UKR	$kernel \in \{\text{Gaussian}, \text{Quartic}, \text{Triweight}\}$
UNN	$k \in \{5, 10, 20, 40\}, \kappa \in \{10, 30, 50\}$
itUKR	$\kappa \in \{30, 50, 70\}, bandwidth \in \{10, 20, 30, 40, 50\}$
Autoenc.[1]	$\lambda \in \{0.0, 0.2, 0.5, 1.0, 1.5, 2.5, 5.0\}$

Tab. A.2: Wertebereiche der Parameter der nichtkonvexen Verfahren

[1] Aus Laufzeitgründen wurde der Autoencoder bei CNS und ORL nur mit $\lambda = 0.0$ ausgeführt. Es wurde $\lambda = 0.0$ gewählt, denn bei den anderen Datensätzen besaß der Autoencoder beim EKS sehr häufig NaN-Werte bei einem größerem λ, da die latenten Punkte nahezu auf eine Stelle gehäuft wurden.

B Inhalt der beigelegten DVD

Die der Arbeit beigelegte DVD enthält das implementierte Rahmengerüst für die Ausführung des experimentellen Vergleichs als Eclipse-Python-Projekt, Installations- und Ausführungshinweise für dieses sowie die verwendeten Matlab-Toolboxen. Außerdem sind die berechneten Metrikwerte aller Experimente und die vorliegende Arbeit enthalten.

+ Ausgaben (.csv- und .ods-Dateien mit Metrikwerten)
+ DRComparison (Eclipse-Python-Projekt)
+ Matlab-Toolboxen (Implementierungen der Verfahren)
– Masterarbeit_AlmuthMeier.pdf (das vorliegende Dokument)
– README.txt (Installations- und Ausführungshinweise)
– requirements.txt (Python-Pakete für das Python-Projekt)

Literaturverzeichnis

[Alp10] Ethem Alpaydin. *Introduction to Machine Learning*. Hrsg. von Thomas Dietterich. Zweite Auflage. Adaptive Computation and Machine Learning. MIT Press, 2010 (siehe S. 2, 8, 19, 21, 52).

[BAM12] Juan M. Banda, Rafal A. Angryk und Petrus C. Martens. „Quantitative Comparison of Linear and Non-linear Dimensionality Reduction Techniques for Solar Image Archives.“ In: *FLAIRS Conference*. 2012 (siehe S. 3).

[Bel61] Richard Ernest Bellman. *Adaptive Control Processes*. Princeton University Press. 1961 (siehe S. 2).

[BG05] Ingwer Borg und Patrick J.F. Groenen. *Modern Multidimensional Scaling*. Springer-Verlag New York, 2005 (siehe S. 19, 34).

[BG88] Marcel Berger und Bernard Gostiaux. *Differential Geometry: Manifolds, Curves, and Surfaces*. Bd. 115. Graduate Texts in Mathematics. Springer New York, 1988 (siehe S. 33).

[BGR94] Victor R. Basili, Caldiera Gianluigi und H. Dieter Rombach. „Goal question metric paradigm“. In: *Encyclopedia of Software Engineering – 2 Volume Set* (1994), S. 528–532 (siehe S. 61).

[Bis95] Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Clarendon Press, 1995 (siehe S. 52).

[BN03] Mikhail Belkin und Partha Niyogi. „Laplacian eigenmaps for Dimensionality Reduction and Data Representation“. In: *Neural computation* 15.6 (2003), S. 1373–1396 (siehe S. 30, 31).

[Bra03] Matthew Brand. „Charting a Manifold“. In: *Advances in Neural Information Processing Systems 15*. Hrsg. von S. Becker, S. Thrun und K. Obermayer. MIT Press, 2003, S. 985–992 (siehe S. 44).

[BSW98] Christopher Bishop, Markus Svensén und Christopher K.I. Williams. „GTM: The Generative Topographic Mapping“. In: *Neural Computation* 10.1 (1998), S. 215–234 (siehe S. 52).

[CC00] Trevor F. Cox und Michael A. A. Cox. *Multidimensional Scaling*. Zweite Auflage. CRC press, 2000 (siehe S. 34, 35, 37).

[CL06] Ronald R. Coifman und Stéphane Lafon. „Diffusion maps“. In: *Applied and Computational Harmonic Analysis* 21.1 (2006), S. 5 –30 (siehe S. 26).

[Coi+05] R. R. Coifman u. a. „Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps“. In: *Proceedings of the National Academy of Sciences of the United States of America* 102.21 (2005), S. 7426–7431 (siehe S. 26).

[CP10] Miguel A Carreira-Perpinán. „The Elastic Embedding Algorithm for Dimensionality Reduction.“ In: *ICML*. Bd. 10. 2010, S. 167–174 (siehe S. 3, 46, 60).

[CPL10] Miguel A. Carreira-Perpinán und Zhengdong Lu. „Parametric dimensionality reduction by unsupervised regression“. In: *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE. 2010, S. 1895–1902 (siehe S. 15).

[DC92] David DeMers und Garrison W. Cottrell. „Non-Linear Dimensionality Reduction“. In: *Advances in Neural Information Processing Systems 5*. Hrsg. von S. J. Hanson, J. D. Cowan und C. L. Giles. Morgan-Kaufmann, 1992, S. 580–587 (siehe S. 53).

[DG03] David L. Donoho und Carrie Grimes. „Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data“. In: *Proceedings of the National Academy of Sciences* 100.10 (2003), S. 5591–5596 (siehe S. 32, 33).

[DH95] Pierre Demartines und Jeanny Hérault. „CCA: ‘Curvilinear component analysis’“. In: *15° Colloque sur le traitement du signal et des images, FRA, 1995*. GRETSI, Groupe d’Etudes du Traitement du Signal et des Images. 1995 (siehe S. 73).

[DKŽ13] Gintautas Dzemyda, Olga Kurasova und Julius Žilinskas. *Multidimensional Data Visualization: Methods and Applications*. Bd. 75. Springer Optimization and Its Applications. Springer Science+Business Media, 2013 (siehe S. 12).

[DLR77] Arthur P. Dempster, Nan M. Laird und Donald B. Rubin. „Maximum Likelihood from Incomplete Data via the EM Algorithm“. In: *Journal of the Royal Statistical Society. Series B (Methodological)* 39.1 (1977), S. 1–38 (siehe S. 38).

[EHH12] Daniel Engel, Lars Hüttenberger und Bernd Hamann. „A Survey of Dimension Reduction Methods for High-dimensional Data Analysis and Visualization“. In: *Visualization of Large and Unstructured Data Sets: Applications in Geospatial Planning, Modeling and Engineering - Proceedings of IRTG 1131 Workshop 2011*. Hrsg. von Christoph Garth, Ariane Middel und Hans Hagen. Bd. 27. OpenAccess Series in Informatics (OASIcs). Dag-

stuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2012, S. 135–149 (siehe S. 12).

[Eve84] Brian Everitt. *An Introduction to Latent Variable Models*. Chapman und Hall, 1984 (siehe S. 37–40).

[FD+14] Manuel Fernández-Delgado u. a. „Do we need hundreds of classifiers to solve real world classification problems?“ In: *Journal of Machine Learning Research* 15.1 (2014), S. 3133–3181 (siehe S. 75, 105).

[For13] Otto Forster. *Analysis 2: Differentialrechnung im IRn, gewöhnliche Differentialgleichungen*. Grundkurs Mathematik. Springer Fachmedien Wiesbaden, 2013 (siehe S. 33).

[GH15] Andrej Gisbrecht und Barbara Hammer. „Data visualization by nonlinear dimensionality reduction“. In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 5.2 (2015), S. 51–73 (siehe S. 3, 12, 13, 35, 36, 59, 66, 67).

[GH97] Zoubin Ghahramani und Geoffrey E. Hinton. *The EM Algorithm for Mixtures of Factor Analyzers*. Technical Report CRG-TR-96-1. Department of Computer Science, University of Toronto, 1997 (siehe S. 39, 40, 42).

[GM12] Bernd Gärtner und Jiří Matoušek. *Approximation Algorithms and Semidefinite Programming*. Springer Berlin Heidelberg, 2012 (siehe S. 59).

[GW12] Xinbo Gao und Xiumei Wang. „Dimensionality reduction with latent variable model“. In: *Frontiers of Electrical and Electronic Engineering* 7.1 (2012), S. 116–126 (siehe S. 38, 41).

[Hal+12] Yoni Halpern u. a. „A comparison of dimensionality reduction techniques for unstructured clinical text“. In: *ICML 2012 workshop on clinical data analysis*. 2012 (siehe S. 3).

[Ham+03] Jihun Ham u. a. *A kernel view of the dimensionality reduction of manifolds*. Techn. Ber. TR-110. Max-Planck-Institut für biologische Kybernetik, 2003 (siehe S. 54).

[HDR97] Geoffrey E. Hinton, Peter Dayan und Michael Revow. „Modeling the Manifolds of Images of Handwritten Digits“. In: *IEEE transactions on Neural Networks* 8.1 (1997), S. 65–74 (siehe S. 60).

[He+05] Xiaofei He u. a. „Neighborhood Preserving Embedding“. In: *Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05)*. Bd. 2. IEEE. 2005, S. 1208–1213 (siehe S. 29, 54).

[Hen+] Sebastian Henschel u. a. *Machine Learning Data Set Repository*. Letzter Zugriff am 14.11.2016. URL: http://mldata.org/repository/data/viewslug/central-nervous-system/ (siehe S. 69).

[Hiv] *Performance Prediction Challenge WCCI 2006*. Letzter Zugriff am 14.11.2016. URL: http://www.modelselect.inf.ethz.ch/datasets.php (siehe S. 69).

[HN03] Xiaofei He und Partha Niyogi. „Locality Preserving Projections“. In: *Advances in Neural Information Processing Systems 16*. Bd. 16. Vancouver, Canada, 2003 (siehe S. 31, 54).

[Hot33] Harold Hotelling. „Analysis of a complex of statistical variables into principal components.“ In: *Journal of Educational Psychology* 24.6 (1933), S. 417 (siehe S. 16).

[HR02] Geoffrey E. Hinton und Sam T. Roweis. „Stochastic Neighbor Embedding“. In: *Advances in Neural Information Processing Systems 15*. Hrsg. von S. Becker, S. Thrun und K. Obermayer. MIT Press, 2002, S. 857–864 (siehe S. 44, 45).

[HS06] Geoffrey E. Hinton und Ruslan R. Salakhutdinov. „Reducing the Dimensionality of Data with Neural Networks“. In: *Science* 313.5786 (2006), S. 504–507 (siehe S. 53).

[HSS08] Thomas Hofmann, Bernhard Schölkopf und Alexander J. Smola. „Kernel methods in machine learning“. In: *The Annals of Statistics* 36.3 (2008), S. 1171–1220 (siehe S. 22, 23).

[HTF09] Trevor Hastie, Robert Tibshirani und Jerome Friedman. *The Elements of Statistical Learning*. Zweite Auflage. Springer New York, 2009 (siehe S. 2, 65).

[Ize13] Alan Julian Izenman. *Modern Multivariate Statistical Techniques*. Hrsg. von George Casella, Stephen Fienberg und Ingram Olkin. Springer Science+Business Media New York, 2013 (siehe S. 8, 12).

[Jol02] I. T. Jolliffe. *Principal Component Analysis*. Springer Series in Statistics. Springer New York, 2002 (siehe S. 16, 18, 20).

[Jör67] Karl G. Jöreskog. „Some contributions to maximum likelihood factor analysis“. In: *Psychometrika* 32.4 (1967), S. 443–482 (siehe S. 40).

[Kla07] Stefan Klanke. „Learning Manifolds with the Parametrized Self-Organizing Map and Unsupervised Kernel Regression.“ Diss. Bielefeld University, 2007 (siehe S. 25, 35, 48).

[Kra+16] Oliver Kramer u. a. *Machine Learning in Evolution Strategies*. Bd. 20. Springer, 2016 (siehe S. 19, 54, 65, 68, 114).

[Kra13] Oliver Kramer. *Dimensionality Reduction with Unsupervised Nearest Neighbors*. Bd. 51. Intelligent Systems Reference Library. Springer, 2013 (siehe S. 2, 3, 22, 48, 49, 54, 114).

[Kra15] Oliver Kramer. „Unsupervised nearest neighbor regression for dimensionality reduction“. In: *Springer Soft Computing* 19.6 (2015), S. 1647–1661 (siehe S. 51, 59, 73).

[Kru64a] Joseph B. Kruskal. „Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis“. In: *Psychometrika* 29.1 (1964), S. 1–27 (siehe S. 36).

[Kru64b] Joseph B. Kruskal. „Nonmetric multidimensional scaling: A numerical method“. In: *Psychometrika* 29.2 (1964), S. 115–129 (siehe S. 36).

[Lan+04] Gert R. G. Lanckriet u. a. „Learning the Kernel Matrix with Semidefinite Programming“. In: *Journal of Machine Learning Research* 5 (2004), S. 27–72 (siehe S. 25).

[Law03] Neil D. Lawrence. „Gaussian Process Latent Variable Models for Visualisation of High Dimensional Data“. In: *Advances in Neural Information Processing Systems 16*. Hrsg. von S. Thrun, L. K. Saul und B. Schölkopf. MIT Press, 2003, S. 329–336 (siehe S. 41).

[Law05] Neil Lawrence. „Probabilistic Non-linear Principal Component Analysis with Gaussian Process Latent Variable Models“. In: *Journal of Machine Learning Research* 6 (2005), S. 1783–1816 (siehe S. 59).

[LB05] Elizaveta Levina und Peter J. Bickel. „Maximum Likelihood Estimation of Intrinsic Dimension“. In: *Advances in Neural Information Processing Systems 17*. Hrsg. von L. K. Saul, Y. Weiss und L. Bottou. MIT Press, 2005, S. 777–784 (siehe S. 11).

[Lic] M. Lichman. *UCI Machine Learning Repository*. Letzter Zugriff am 14.11.2016. URL: `http://archive.ics.uci.edu/ml` (siehe S. 69).

[LK14] Daniel Lückehe und Oliver Kramer. „Leaving Local Optima in Unsupervised Kernel Regression“. In: *Artificial Neural Networks and Machine Learning - ICANN 2014 - 24th International Conference on Artificial Neural Networks, Hamburg, Germany, September 15-19, 2014. Proceedings*. 2014, S. 137–144 (siehe S. 51).

[LK15] Daniel Lückehe und Oliver Kramer. „Alternating Optimization of Unsupervised Regression with Evolutionary Embeddings“. In: *Applications of Evolutionary Computation - 18th European Conference, EvoApplications 2015, Copenhagen, Denmark, April 8-10, 2015, Proceedings*. 2015, S. 471–480 (siehe S. 51).

[LMS12] Joshua M. Lewis, Laurens van der Maaten und Virginia R. de Sa. „A Behavioral Investigation of Dimensionality Reduction“. In: *Proceedings of the 34th Annual Meeting of the Cognitive Science Society*. 2012 (siehe S. 63).

[Lue+11] Wouter Lueks u. a. „How to Evaluate Dimensionality Reduction? - Improving the Co-ranking Matrix“. In: *CoRR* abs/1110.3917 (2011) (siehe S. 63, 64, 68).

[LV07] John A. Lee und Michel Verleysen. *Nonlinear Dimensionality Reduction*. Springer Science & Business Media, 2007 (siehe S. 9, 11, 12, 30).

[LV09] John A. Lee und Michel Verleysen. „Quality assessment of dimensionality reduction: Rank-based criteria“. In: *Neurocomputing* 72.7 (2009), S. 1431–1443 (siehe S. 63–65, 74).

[Lüc15] Daniel Lückehe. *Hybride Optimierung für Dimensionsreduktion: Unüberwachte Regression mit Gradientenabstieg und evolutionären Algorithmen*. Springer Vieweg, 2015 (siehe S. 51).

[Maa] Laurens van der Maaten. *Matlab Toolbox for Dimensionality Reduction*. Letzter Zugriff am 28.10.2016. URL: https://lvdmaaten.github.io/drtoolbox/ (siehe S. 9, 69, 71, 73).

[Maa14] Laurens van der Maaten. „Accelerating t-SNE using Tree-Based Algorithms“. In: *Journal of Machine Learning Research* 15 (2014), S. 3221–3245 (siehe S. 46).

[Mar09] Stephen Marsland. *Machine Learning: An Algorithmic Perspective*. Hrsg. von Ralf Herbrich und Thore Graepel. CRC Press/Taylor & Francis, 2009 (siehe S. 2, 15, 16, 22, 28, 36, 39).

[Mei+05] Peter Meinicke u. a. „Principal surfaces from unsupervised kernel regression“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27.9 (2005), S. 1379–1391 (siehe S. 47–49, 59).

[MH08] Laurens van der Maaten und Geoffrey Hinton. „Visualizing data using t-SNE“. In: *Journal of Machine Learning Research* 9.2579-2605 (2008), S. 85 (siehe S. 3, 44–46, 54, 59).

[MHP11] Peter Mysling, Søren Hauberg und Kim Steenstrup Pedersen. „An empirical study on the performance of spectral manifold learning techniques“. In: *Artificial Neural Networks and Machine Learning–ICANN 2011*. Springer, 2011, S. 347–354 (siehe S. 3).

[Mok+13] Bassam Mokbel u. a. „Visualizing the quality of dimensionality reduction“. In: *Neurocomputing* 112 (2013), S. 109–123 (siehe S. 64).

[MPH07] Laurens van der Maaten, E.O. Postma und H.J. van den Herik. „Dimensionality Reduction: A Comparative Review“. In: *Journal of Machine Learning Research* 10.1 (2007) (siehe S. 13).

[MPH09] Laurens van der Maaten, E.O. Postma und H.J. van den Herik. *Dimensionality Reduction: A Comparative Review*. Technical Report TiCC-TR 2009-005. Tilburg University, 2009 (siehe S. 3, 11–14, 18–21, 23–27, 30, 32, 35, 36, 41–44, 53, 54, 58, 59, 64, 71, 81, 82, 107).

[NS03] Matti Niskanen und Olli Silvén. „Comparison of dimensionality reduction methods for wood surface inspection“. In: *Quality Control by Artificial Vision*. International Society for Optics und Photonics. 2003, S. 178–188 (siehe S. 3, 67).

[Orl] *The Database of Faces*. Letzter Zugriff am 14.11.2016. URL: http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html (siehe S. 69).

[Pan+05] Yanwei Pang u. a. „Neighborhood Preserving Projections (NPP): A Novel Linear Dimension Reduction Method“. In: *International Conference on Intelligent Computing*. Springer. 2005, S. 117–125 (siehe S. 29).

[Pea01] Karl Pearson. „On lines and planes of closest fit to systems of points in space“. In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2.11 (1901), S. 559–572 (siehe S. 16).

[Ped+11] F. Pedregosa u. a. „Scikit-learn: Machine Learning in Python“. In: *Journal of Machine Learning Research* 12 (2011), S. 2825–2830 (siehe S. 69, 73).

[PG10] Stefan Pintilie und Ali Ghodsi. „Conformal Mapping by Computationally Efficient Methods.“ In: *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI-10)*. 2010 (siehe S. 59).

[Qia+13] Hong Qiao u. a. „An Explicit Nonlinear Mapping for Manifold Learning“. In: *IEEE transactions on cybernetics* 43.1 (2013), S. 51–63 (siehe S. 29).

[QY04] Michael Quist und Golan Yona. „Distributional scaling: An Algorithm for Structure-Preserving Embedding of Metric and Nonmetric Spaces“. In: *Journal of Machine Learning Research* 5 (2004), S. 399–420 (siehe S. 35–37).

[RL15] Bastian Rieck und Heike Leitte. „Comparing Dimensionality Reduction Methods Using Data Descriptor Landscapes“. In: *Symposium on Visualization in Data Science (VDS) at IEEE VIS 2015*. Chicago, Illinois, Okt. 2015 (siehe S. 68).

[Row97] Sam T. Roweis. „EM Algorithms for PCA and SPCA“. In: *Advances in Neural Information Processing Systems 10*. Hrsg. von M. I. Jordan, M. J. Kearns und S. A. Solla. MIT Press, 1997, S. 626–632 (siehe S. 41, 58, 59).

[RS00] Sam T. Roweis und Lawrence K. Saul. „Nonlinear Dimensionality Reduction by Locally Linear Embedding“. In: *Science* 290.5500 (2000), S. 2323–2326 (siehe S. 27, 28).

[Sam69] J. W. Sammon. „A Nonlinear Mapping for Data Structure Analysis“. In: *IEEE Trans. Comput.* 18.5 (1969), S. 401–409 (siehe S. 36).

[San08] Guido Sanguinetti. „Dimensionality reduction of clustered data sets“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30.3 (2008), S. 535–540 (siehe S. 66).

[SIL15] Guzman Santafe, Iñaki Inza und Jose A. Lozano. „Dealing with the evaluation of supervised classification algorithms“. In: *Artificial Intelligence Review* 44.4 (2015), S. 467–508 (siehe S. 108).

[Spe04] Charles Spearman. „‘General Intelligence,’ objectively determined and measured“. In: *The American Journal of Psychology* 15.2 (1904), S. 201–292 (siehe S. 39).

[SS05] Fei Sha und Lawrence K. Saul. „Analysis and Extension of Spectral Methods for Nonlinear Dimensionality Reduction“. In: *Proceedings of the 22nd International Conference on Machine learning*. ACM. 2005, S. 784–791 (siehe S. 32, 59).

[SSM98] Bernhard Schölkopf, Alexander Smola und Klaus-Robert Müller. „Nonlinear Component Analysis as a Kernel Eigenvalue Problem“. In: *Neural Computation* 10.5 (1998), S. 1299–1319 (siehe S. 22).

[ST03] Vin D. Silva und Joshua B. Tenenbaum. „Global Versus Local Methods in Nonlinear Dimensionality Reduction“. In: *Advances in Neural Information Processing Systems 15*. Hrsg. von S. Becker, S. Thrun und K. Obermayer. MIT Press, 2003, S. 721–728 (siehe S. 54).

[Sve98] Johan Fredrik Markus Svensén. „GTM: The Generative Topographic Mapping“. Dissertation. Aston University, 1998 (siehe S. 52).

[TDSL00] Joshua B. Tenenbaum, Vin De Silva und John C. Langford. „A global geometric framework for nonlinear dimensionality reduction“. In: *science* 290.5500 (2000), S. 2319–2323 (siehe S. 20).

[TKR08] Ameet Talwalkar, Sanjiv Kumar und Henry Rowley. „Large-scale manifold learning“. In: *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE. 2008, S. 1–8 (siehe S. 12).

[Tor52] Warren S. Torgerson. „Multidimensional scaling: I. Theory and method“. In: *Psychometrika* 17.4 (1952), S. 401–419 (siehe S. 18).

[Tor58] Warren S. Torgerson. *Theory and Methods of Scaling.* New York: Wiley. 1958 (siehe S. 18).

[TR02] Yee W. Teh und Sam T. Roweis. „Automatic Alignment of Local Representations“. In: *Advances in Neural Information Processing Systems 15.* Hrsg. von S. Becker, S. Thrun und K. Obermayer. MIT Press, 2002, S. 865–872 (siehe S. 42–44).

[TSK06] Pang-Ning Tan, Michael Steinbach und Vipin Kumar. *Introduction to Data Mining.* Pearson/Addison Wesley, 2006 (siehe S. 16, 36, 39).

[VCP12] M. Vladymyrov und M. Á. Carreira-Perpiñán. „Partial-Hessian strategies for fast learning of nonlinear embeddings“. In: *Proceedings of the 29th International Conference on Machine Learning* (2012), S. 345 –352 (siehe S. 73).

[Ven+10] Jarkko Venna u. a. „Information retrieval perspective to nonlinear dimensionality reduction for data visualization“. In: *Journal of Machine Learning Research* 11.Feb (2010), S. 451–490 (siehe S. 64).

[Ver04] Jakob Verbeek. „Mixture models for clustering and dimension reduction“. Diss. Universiteit van Amsterdam, 2004 (siehe S. 38).

[VK01] Jarkko Venna und Samuel Kaski. „Neighborhood preservation in nonlinear projection methods: An experimental study“. In: *Artificial Neural Networks—ICANN 2001.* Springer, 2001, S. 485–491 (siehe S. 64).

[VK06] Jarkko Venna und Samuel Kaski. „Visualizing gene interaction graphs with local multidimensional scaling.“ In: *ESANN 2006 proceedings - European Symposium on Artificial Neural Networks Bruges (Belgium).* Bd. 6. 2006, S. 557–562 (siehe S. 64).

[WC05] Sitao Wu und Tommy W. S. Chow. „PRSOM: a new visualization method by hybridizing multidimensional scaling and self-organizing map“. In: *IEEE Transactions on Neural Networks* 16.6 (2005), S. 1362–1380 (siehe S. 54).

[Wil02] Christopher K. I. Williams. „On a connection between Kernel PCA and Metric Multidimensional Scaling“. In: *Machine Learning* 46.1-3 (2002), S. 11–19 (siehe S. 19, 20, 35–37).

[WM97] David H. Wolpert und William G. Macready. „No free lunch theorems for optimization“. In: *IEEE transactions on evolutionary computation* 1.1 (1997), S. 67–82 (siehe S. 111).

[WMS00] P. Wu, B. S. Manjunath und H. D. Shin. „Dimensionality Reduction for Image Retrieval“. In: *Proceedings 2000 International Conference on Image Processing*. Bd. 3. IEEE. 2000, S. 726–729 (siehe S. 1).

[XSB06] Lin Xiao, Jun Sun und Stephen Boyd. „A duality view of spectral methods for dimensionality reduction“. In: *Proceedings of the 23rd international conference on Machine learning*. ACM. 2006, S. 1041–1048 (siehe S. 15).

[Zha+07] Tianhao Zhang u. a. „Linear local tangent space alignment and application to face recognition“. In: *Neurocomputing* 70.7 (2007), S. 1547–1553 (siehe S. 34, 58).

[Zha+09] Tianhao Zhang u. a. „Patch Alignment for Dimensionality Reduction“. In: *IEEE Transactions on Knowledge and Data Engineering* 21.9 (2009), S. 1299–1313 (siehe S. 31, 33).

[ZRZ12] Peng Zhang, Yuanyuan Ren und Bo Zhang. „A new embedding quality assessment method for manifold learning“. In: *Neurocomputing* 97 (2012), S. 251–266 (siehe S. 68).

[ZW06] Zhenyue Zhang und Jing Wang. „MLLE: Modified Locally Linear Embedding Using Multiple Weights“. In: *Advances in Neural Information Processing Systems 19*. Hrsg. von B. Schölkopf, J. C. Platt und T. Hoffman. Bd. 19. MIT Press, 2006, S. 1593–1600 (siehe S. 28).

[ZZ04] Zhen-yue Zhang und Hong-yuan Zha. „Principal Manifolds and Nonlinear Dimensionality Reduction via Tangent Space Alignment“. In: *Journal of Shanghai University (English Edition)* 8.4 (2004), S. 406–424 (siehe S. 33).

[El] Laurent El Ghaoui. *Optimization models and applications*. `https://inst.eecs.berkeley.edu/~ee127a/book/login/index.html`. Letzter Zugriff am 10.10.2016 (siehe S. 15).

Index

Anteil am minimalen Fehler, 105
artificial neural network, *siehe* künstliches neuronales Netz
Autoencoder, 53

beobachtete Variable, 37

CE, *siehe* Conformal Eigenmaps
Classical Scaling, 18
clustering, *siehe* Gruppierung
Co-Ranking Matrix, 63
Conformal Eigenmaps, 32
CS, *siehe* Classical Scaling
Curvilinear Component Analysis, 73

Datenraum, 2, 7
Diffusion Maps, 26
DM, *siehe* Diffusion Maps

EE, *siehe* Elastic Embedding
Eigenwertzerlegung, 14
Elastic Embedding, 46
EM-Algorithmus, *siehe* Expectation-Maximization
Expectation-Maximization, 38
Experiment, 74

FA, *siehe* Factor Analysis
Factor Analysis, 39
Familie, 2, 13
feature extraction, *siehe* Merkmalsextraktion
feature selection, *siehe* Merkmalsauswahl
Fluch der Dimensionen, 2

Gaussian Process Latent Variable Model, 41
Generative Topographic Mapping, 52
GPLVM, *siehe* Gaussian Process Latent Variable Model
Gradientenabstieg, 35
Gram-Matrix, 19, 23

Gruppierung, 2

Hesse-Matrix, 33
Hessian Eigenmaps, 32
Hessian Locally Linear Embedding, 32
HLLE, *siehe* Hessian Locally Linear Embedding
hybrides Dimensionsreduktionsverfahren, 54, 114

intrinsische Dimensionalität, 11
intrinsische Struktur, *siehe* Mannigfaltigkeit
Isomap, *siehe* Isometric Feature Mapping
Isometric Feature Mapping, 20
iterative UKR, 51
itUKR, *siehe* iterative UKR

k-nächste-Nachbarn, 49
künstliches neuronales Netz, 52
Kern, 22
Kern-Matrix, 23
Kernel PCA, 22
Kernel Regression, 48
KNN-Klassifikation, 67
KNN-Regression, 47, 49, 68
KPCA, *siehe* Kernel PCA
Kullback-Leibler-Divergenz, 45

landmark-Variante, 54
Laplace-Matrix, 30
Laplacian Eigenmaps, 30
Laplacianfaces, 31
latent variable model, 7, 37, 91
latente Einbettung, 7
latente Variable, 7, 37
latenter Punkt, 7
latenter Raum, 7
LE, *siehe* Laplacian Eigenmaps
Least-Squares Scaling, 36
LLC, *siehe* Locally Linear Coordination
LLE, *siehe* Locally Linear Embedding
Local Tangent Space Alignment, 33
Locality Preserving Projections, 31
Locally Linear Coordination, 42
Locally Linear Embedding, 27
LPP, *siehe* Locality Preserving Projections
LTSA, *siehe* Local Tangent Space Alignment
LVM, *siehe* latent variable model

manifold, *siehe* Mannigfaltigkeit
Manifold Charting, 44
manifold learner, 12
Mannigfaltigkeit, 9
Maximum Variance Unfolding, 24
Maximum-Likelihood-Schätzer, 11,

69
MC, *siehe* Manifold Charting
MDS, *siehe* Multidimensional Scaling
Merkmalsabbildung, 22
Merkmalsauswahl, 2
Merkmalsextraktion, 2
Merkmalsraum, 22
metric MDS, 35
MFA, *siehe* mixture of factor analyzers
mixture model, 37, 91
mixture of factor analyzers, 42
MLLE, *siehe* Modified Locally Linear Embedding
MM, *siehe* mixture model
mMDS, *siehe* metric MDS
Modified Locally Linear Embedding, 28
Multidimensional Scaling, 34
Muster, 2
MVU, *siehe* Maximum Variance Unfolding

Nadaraya-Watson-Schätzer, 49
Neighborhood Preserving Embedding, 29
Neighborhood Preserving Projections, 29
nMDS, *siehe* nonmetric MDS
no-free-lunch, 111
nonmetric MDS, 36
NPE, *siehe* Neighborhood Preserving Embedding
NPP, *siehe* Neighborhood Preserving Projections

Ordinal Scaling, 37
out-of-sample-Erweiterung, 54, 73

PCA, *siehe* Principal Components Analysis
PPCA, *siehe* Probabilistic PCA
Principal Components Analysis, 16
Principal Coordinate Analysis, 18
Probabilistic PCA, 41
Projektionsmethode, 12

Regression, 47
Regressionsfunktion, 47

Sammon Mapping, 36
self-organizing map, 52
Semidefinite Embedding, 24
semidefinite Programmierung, 25, 32
SM, *siehe* Sammon Mapping
SNE, *siehe* Stochastic Neighbor Embedding
spektrale Verfahren, 15
Spektralzerlegung, *siehe* Eigenwertzerlegung

Stochastic Neighbor Embedding, 44
submanifold learner, 54, 114

t-Distributed SNE, 45
t-SNE, *siehe* t-Distributed SNE
Tangentialraum, 33

UKR, *siehe* Unsupervised Kernel Regression
UNN, *siehe* Unsupervised Nearest Neighbors
Unsupervised Kernel Regression, 48
Unsupervised Nearest Neighbors, 49

verallgemeinertes Eigenwertproblem, 31, 32, 43

Selbstständigkeitserklärung

Hiermit versichere ich, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Außerdem versichere ich, dass ich die allgemeinen Prinzipien wissenschaftlicher Arbeit und Veröffentlichung, wie sie in den Leitlinien guter wissenschaftlicher Praxis der Carl von Ossietzky Universität Oldenburg festgelegt sind, befolgt habe.