Diplomarbeit

Einbettung von Zeitreihen nach Takens' Theorem

angefertigt am
Institut für Numerische Simulation

vorgelegt der
Mathematisch-Naturwissenschaftlichen Fakultät der
Rheinischen Friedrich-Wilhelms-Universität Bonn

März 2010

von
Bastian Bohn
aus
Traben-Trarbach

Inhaltsverzeichnis

1 Einleitung

In der heutigen Informationsgesellschaft finden sich unzählige Möglichkeiten zur Beschaffung von Daten jeglicher Art. Ob politische Straßenumfragen, Niederschlagsmessungen einer Wetterstation, Erhebungen der Anzahl der Grippefälle in einem Krankenhaus, das Zählen der Besucher einer Website oder das Notieren des Kursstandes einer Aktie, die resultierenden Daten haben zwei Gemeinsamkeiten:

- Sie sind von verschiedenen Umständen (u.a. dem Zeitpunkt der Datenbeschaffung) abhängig.
- Sie sind ohne entsprechende Verarbeitung meist nicht von Bedeutung.

Die vorliegenden Daten sind oftmals zu komplex und umfangreich, um sie von Hand zu analysieren. Dies begründet die Vielfalt an *maschinellen Lernmethoden*, die heutzutage existieren. Diese dienen dazu, Muster und Zusammenhänge in den vorliegenden Rohdaten zu erkennen, und bieten somit die Möglichkeit einer detaillierten Analyse. Beispiele für solche Methoden sind *Support-Vektor-Maschinen* [SS02], *Neuronale Netze* [KKKW95] oder auch allgemeine *Clustering-Methoden* [BMDG05]. Den gesamten Prozess des Erhebens der Daten, der Verarbeitung mittels maschineller Lernmethoden sowie die anschließende Auswertung der Ergebnisse bezeichnet man als *Data Mining*.
Die Umstände, unter denen die Daten erhoben wurden, können die Ergebnisse der Analyse maßgeblich beeinflussen: Die Anzahl der Grippefälle ist saisonbedingt und Niederschlagsmessungen werden im europäischen Raum nicht dieselben Ergebnisse liefern wie am Äquator. Wünschenswert ist es allerdings, dass Datenerhebungen unter gleichen Umständen auch die gleichen Ergebnisse liefern und die Messungen somit deterministisch sind.
In dieser Arbeit beschäftigen wir uns mit der Analyse von *Zeitreihen*. Zeitreihendaten zeichnen sich dadurch aus, dass sie chronologisch angeordnet werden können. Führt man die oben genannten Messungen zu verschiedenen Zeitpunkten durch, so können die resultierenden Daten als Zeitreihe aufgefasst werden. Im Rahmen dieser Arbeit untersuchen wir ausschließlich Zeitreihen mit Werten in $\mathbb{R}$, die meisten Beobachtungen und Sätze lassen sich jedoch auf den mehrdimensionalen Fall verallgemeinern.
Das Hauptaufgabengebiet der Zeitreihenanalyse ist die Vorhersage einer vorliegenden Zeitreihe, also die Prognose der zukünftige Entwicklung. Einsatzgebiete sind beispielsweise Niederschlagsvorhersagen, Aktienkursprognosen oder Epidemiewarnsysteme. Es stellt sich nun die Frage, ob die entsprechenden Zeitreihen überhaupt genügend Informationen beinhalten, um eine solche Vorhersage zu ermöglichen, da beispielsweise der Niederschlag

von vielen Faktoren abhängig ist. Dass dennoch eine Zeitreihenanalyse mittels bereits bekannten maschinellen Lernmethoden möglich ist, zeigt ein von Floris Takens in [Tak81] entwickeltes Theorem.
Die Nutzung von auf das entsprechende Anwendungsgebiet angepassten Vorhersagemethoden ist dennoch sinnvoll, da die allgemeine Zeitreihenanalyse, wie sie hier vorgestellt wird, nur unter speziellen Voraussetzungen wie beispielsweise der *Stationärität* der Zeitreihe möglich ist.
In dieser Arbeit beschäftigen wir uns insbesondere mit den strukturellen Voraussetzungen, welche eine Zeitreihe $(x_t)_t$ erfüllen muss, damit eine Anwendung von Takens' Theorem möglich ist. Aus dem in [Tak81] vorgestellten Prinzip der *Delay-Einbettung* resultieren anschließend Vektoren im reellen Vektorraum $\mathbb{R}^{2d+1}$, wobei d die *Boxcounting*-Dimension des *Attraktors* des zugrundeliegenden Prozesses ist. Diese Vektoren haben die Gestalt

$$\mathbf{x}_t = (x_t, x_{t-1}, x_{t-2}, \ldots, x_{t-2d})^T.$$

Das Ziel der Vorhersage ist, einem Vektor $\mathbf{x}_t$ eindeutig den Zeitreihenwert x_{t+1} zuordnen zu können. Zu diesem Zweck verwenden wir den in [Gar04] entworfenen Algorithmus als Lernmethode und minimieren das Funktional

$$\frac{1}{N-(2d+1)} \sum_{t=2d+1}^{N-1} (f(\mathbf{x}_t) - x_{t+1})^2 + \lambda ||f||_*,$$

wobei N die Länge der vorliegenden Zeitreihe, $\lambda \in [0, \infty)$ der sogenannte *Regularisierungsparameter* und $*$ eine bestimmte Norm ist, welche die Glattheit der Funktion f bestimmt. Die Minimierung findet hierbei über dem Raum der stückweise linearen Funktionen auf einem *dünnen Gitter* statt. Der Vorteil dieser Methode gegenüber anderen Lernmethoden ist, dass die Komplexität des Verfahrens nur noch linear in der Anzahl der Zeitreihenpunkte skaliert, obwohl ein nichtlinearer Ansatz vorliegt. Die Dünngitterdiskretisierung sorgt dafür, dass die Komplexität in Bezug auf die Maschenweite h der Diskretisierung nicht mehr $O(h^{(2d+1)})$ wie bei einem vollen Gitter, sondern $O(h \cdot \log(h)^{2d})$ beträgt und der sogenannte *Fluch der Dimension* – siehe [Bel57] – nur noch in sehr geringem Maß auftritt.
An dieser Stelle fassen wir die eigenen Beiträge dieser Arbeit zusammen:

- Wir liefern eine ausführliche Analyse verschiedener Dimensionsschätzer und zweier Schätzer für den sogenannten *Time-Lag*, insbesondere in Hinblick auf eine Eignung in der Praxis. Sämtliche diskutierten Verfahren – insbesondere ein Boxcounting-Schätzer, der sich an [Kru96] orientiert sowie ein Korrelationsdimensionsschätzer, der sich an [The87] orientiert – werden implementiert. Die beiden *Renyi*-Dimensionsschätzer schlagen die Komplexität eines naiven Ansatzes und finden auch in höheren Dimensionen Verwendung. Sämtliche Schätzer werden zu einem Tool zusammengefasst, welches genutzt werden kann, um Zeitreihen zu analysieren und anschließend einzubetten.

- Dass der *Concentration of Measure*-Effekt auch bei der Zeitreihenanalyse auftritt, wird experimentell anhand des Korrelationsdimensionsalgorithmus gezeigt. Es wird eine Erweiterung der bekannten Algorithmen präsentiert, um andere Distanzbegriffe – insbesondere Bregman-Divergenzen – verwenden und so den Concentration of Measure-Effekt besser kontrollieren zu können.

- Der in [Gar04] implementierte Code wird um die Möglichkeit erweitert, eine H^1_{mix}-Regularisierung durchzuführen, da diese – im Gegensatz zur H^1-Variante – mit der Theorie der reproduzierenden-Kern-Hilbertraum-Regularisierung vereinbar ist. Des Weiteren wird der hierarchische Fehlerindikator in der dimensionsadaptiven Variante des Algorithmus um die Möglichkeit erweitert verschiedene Normen für die Auswertung der Basisfunktionen zu wählen. Für die anderen Fehlerindikatoren wird ein Verfahren vorgestellt, welches mittels einer unteren l_2-Fehlerschranke eine Alternative zur ursprünglichen Herangehensweise darstellt.

- Der im Rahmen von [FG09, Feu10] implementierte Code wird so abgeändert, dass er auf unser Problem anwendbar ist und somit eine ortsadaptive Diskretisierung ermöglicht.

Wir fassen nun kurz den Aufbau der Arbeit zusammen:
In Kapitel 2 finden sich theoretische Grundlagen, die nötig sind, um das Problem der Zeitreihenanalyse zu verstehen. Es werden Definitionen und Erläuterungen zu den relevanten Begriffen der Zeitreihe, der Dissipativität und des Attraktors präsentiert und anhand der Beispiele des Lorenz-Systems und der Henon-Abbildung erklärt. Kapitel 3 beschäftigt sich ausschließlich mit Takens' Theorem und den Voraussetzungen zur Einbettung einer Zeitreihe. Die zeitdiskrete sowie die zeitkontinuierliche Fassung des Theorems werden ausführlich diskutiert. Der Zusammenhang zu anderen bekannten Einbettungssätzen und Theoremen über Approximationen hochdimensionaler Strukturen wird kurz erläutert. In Kapitel 4 setzen wir uns ausführlich mit der praktischen Anwendung von Takens' Theorem auseinander und erklären die Probleme, die insbesondere bei der Dimensionsschätzung und der Wahl des Zeitparameters auftreten. Es werden verschiedene Lösungsvorschläge diskutiert und ausführlich analysiert. Zudem werden die Probleme des euklidischen Abstandsmaß in hochdimensionalen Räumen erläutert und der Zusammenhang zwischen exponentiellen Familien von Wahrscheinlichkeitsmaßen und Bregman-Divergenzen wird vorgestellt. Kapitel 5 beschäftigt mit der in [Gar04] präsentierten Lernmethode und motiviert zunächst das verwendete Zielfunktional, welches minimiert werden soll. Der Zusammenhang zwischen Regularisierung und Hilberträumen mit reproduzierendem Kern wird kurz skizziert. Anschließend wird das aus der Minimierung resultierende Gleichungssystem für eine allgemeine raumbasierte sowie eine spezielle datenbasierte Kern-Basis hergeleitet. Die Verwendung dünner Gitter bei der Diskretisierung wird motiviert und die nötigen Konstruktionen werden erläutert. Zuletzt wird eine dimensionsadaptive Variante des Algorithmus sowie das Prinzip der *ANOVA-Zerlegung* vorgestellt und auf die Möglichkeit einer ortsadaptiven Dis-

kretisierung hingewiesen. In Kapitel 6 werden die implementierten Algorithmen hinsichtlich ihrer Laufzeit und Speicherkomplexität analysiert. Diverse technische Details werden näher erläutert. Kapitel 7 präsentiert die Resultate der Anwendung der Algorithmen auf synthetische sowie reale, wirtschaftsbezogene Datensätze. Zunächst werden experimentelle Konvergenzraten einiger Algorithmen bestimmt und mit den zu erwartenden Raten verglichen. Im Anschluß werden zwei Experimente erläutert, welche Minkowski-Normen sowie Bregman-Divergenzen statt der euklidischen Norm verwenden. Dann findet sich ein Experiment zu den vorgestellten adaptiven Lernverfahren. Zum Abschluß testen wir die Algorithmen anhand zweier bekannter *Benchmark*-Datensätze. In Kapitel 8 findet sich schließlich eine Zusammenfassung sowie ein kurzer Ausblick.

An dieser Stelle möchte ich mich bei allen bedanken, die mich bei der Erstellung dieser Arbeit unterstützt haben.
Als Erstes möchte ich Prof. Dr. Michael Griebel für die Bereitstellung des interessanten Themas sowie viele Ratschläge, Hinweise und Anregungen danken, die mir seit Anbeginn meiner Tätigkeit als studentische Hilfskraft halfen, mich in den Themenbereich einzuarbeiten. Ohne diese Betreuung und die zahlreichen angebotenen Vorlesungen wäre ein erfolgreiches Arbeiten wesentlich schwieriger gewesen und viele Interessen wären nicht geweckt worden.
Bei Priv.-Doz. Dr. Marc Alexander Schweitzer bedanke ich mich für die Übernahme des Zweitgutachtens sowie eine interessante Vorlesung über verallgemeinerte Finite Elemente Methoden, die mir auch sehr für meine Prüfung half.
Dr. Jochen Garcke danke ich für die Bereitstellung seines Codes sowie die Hilfestellungen bei Fragen meinerseits.
Bei Christian Feuersänger möchte ich mich für die vielen hilfreichen Gespräche und guten Hinweise bedanken. Ohne seine zahlreichen Implementierungen und den gut dokumentierten Code hätten auch viele kleine Aufgaben wesentlich mehr Zeit in Anspruch genommen. Neben dem Dünngitter-Code habe ich außerdem sein sehr empfehlenswertes `PGFPLOTS`-Paket verwendet.
Bei Alexander Hullmann und Jens Oettershagen bedanke ich mich für unzählige Gespräche und fachliche sowie nicht-fachliche Diskussionen. Ebenso danke ich den beiden sowie Christian Kuske für die Hilfe bei der Korrektur der Arbeit. Außerdem möchte ich Alexander für seine Geduld danken, die er zur Beantwortung meiner unzähligen Fragen aufbrachte.
Schließlich möchte ich mich bei meinem gesamten Freundeskreis bedanken, ohne den das Studium nicht annähernd so schön und heiter verlaufen wäre. Besonderer Dank gilt auch meiner Familie, die mich sowohl finanziell als auch moralisch unterstützt und gestärkt hat. Zuletzt sei auch meiner Gitarre gedankt, die es in den verbleibenden unstressigen Stunden schaffte, Langeweile gar nicht erst aufkommen zu lassen.

2 Theoretische Grundlagen

In diesem Kapitel wollen wir die für diese Arbeit wichtigen Begriffe definieren, erklären und anhand von Beispielen erläutern. Eine der Hauptintentionen ist – neben der formalen Definition –, dass die Begriffe einfach und intuitiv handhabbar werden sollen. Des Weiteren soll deutlich werden, dass die hier vorgestellten Konzepte auch in der Praxis relevant sind und sich nicht lediglich auf wenige pathologische Fälle beziehen.
Im Folgenden werden Kenntnisse der elementaren Differentialgeometrie vorausgesetzt. Auf die nähere Erläuterung grundlegender Konzepte, wie z.B. C^k-Mannigfaltigkeiten, Kartenwechseln oder Zusammenhängen wird hier bewusst verzichtet, da diese nicht im Vordergrund stehen sollen. Außerdem werden sich Beispiele und praxisrelevante Fälle immer am $\mathbb{R}^n$ orientieren. Zum Nachschlagen der differentialgeometrischen Konzepte sei auf [doC92] verwiesen.

2.1 Der Begriff "Zeitreihe"

Zunächst soll verdeutlicht werden, was der Begriff der *Zeitreihe* zu bedeuten hat. In der praxisrelevanten Anwendung handelt es sich häufig um eine über die Zeit diskret-indizierte Folge $(a_t)_t$ von reellen Werten. Hierbei ist weder klar, ob diese Werte einem zugrundeliegenden Schema folgen, noch ist gesichert, dass sie reproduzierbar sind. Letzteres bezieht sich darauf, ob der Prozess, welcher die Werte erzeugt, deterministisch oder stochastisch ist. Wir befassen uns hier ausschließlich mit deterministischen Prozessen, die in der Praxis maximal ein leichtes stochastisches Rauschen aufweisen. Oftmals stammen die Werte einer Zeitreihe aus der Beobachtung eines zu untersuchenden (z.B. physikalischen) Prozesses. Hierbei wird angenommen, dass dem eigentlichen Prozess eine eventuell mehrdimensionale Struktur zugrunde liegt, die Beobachtungen jedoch als reelle Zahlen notiert werden. Die Annahme eines mehrdimensionalen Prozesses wird dadurch motviert, dass man den Zustand des beobachteten Systems eindeutig durch einen Punkt auf dieser mehrdimensionalen Struktur charakterisieren möchte. Optimal wäre somit eine 1-zu-1 Beziehung zwischen den Punkten auf der Struktur und dem Zustand des Prozesses.

2.1.1 Ein einfaches Beispiel: Das Lorenz-System

Beispiel 2.1: [**Lorenz-System (Einführung)**] Ein Beispiel findet sich im Lorenz-System, das 1962 von Edward Lorenz entwickelt wurde, um Konvektionsströmungen zu beschreiben. Es handelt sich hierbei um ein gekoppeltes System von gewöhnlichen

Differentialgleichungen mit reellen Parametern a, b, c und reellwertigen, von der Zeit abhängigen Funktionen X, Y, Z:

$$\begin{aligned} \dot{X} &= a(Y - X) \\ \dot{Y} &= X(b - Z) - Y \\ \dot{Z} &= XY - cZ \end{aligned} \tag{2.1}$$

Hierbei ist

- X die vertikale Konvektionsgeschwindigkeit,
- Y der Temperaturunterschied zwischen auf- und absteigendem Fluid,
- Z die Abweichung der vertikalen Geschwindigkeit von einer linearen Entwicklung,
- a die *Prandtl*-Zahl,
- b die relative *Rayleigh*-Zahl und
- c ein Maß für die Konvektionszellengeometrie.

Für eine genauere Beschreibung des Modells verweisen wir auf [Lor63]. Die zeitliche Entwicklung der dreidimensionalen Lösungen interpretieren wir hierbei als mehrdimensionalen Prozess. Eine mögliche Beobachtung, bzw. *Observable* $o : \mathbb{R}^3 \to \mathbb{R}$ des Prozesses wäre zum Beispiel $o\left((x_1, x_2, x_3)^T\right) = x_1$. Die resultierende Zeitreihe wäre somit

$$\left(o\left((X(t), Y(t), Z(t))^T\right)\right)_t = \left(X(t)\right)_t ,$$

wobei noch keine Aussage über die Indizierung stattgefunden hat. Da es in der Praxis meist nicht möglich ist, zeitkontinuierliche Messergebnisse zu liefern, muss man den Prozess zu bestimmten Zeitpunkten $(t_0, t_1, t_2, t_3, \dots)$ betrachten und die Observable auswerten. Denselben Effekt kann man durch diskretes Sampling des zeitkontinuierlichen Prozesses erreichen. Dies führt zu folgender Zeitreihe:

$$\left(X(t_i)\right)_{t_i} ; \quad i \in \mathbb{N}$$

Oftmals sind die Zeitpunkte t_i äquidistant gewählt. Diese zusätzliche Einschränkung ist eine Voraussetzung, um Takens' Theorem 3.1 anwenden zu können.
Bis jetzt haben wir noch keinerlei Forderungen an den Prozess gestellt oder ihn genauer definiert. Ebenfalls könnte man jede beliebige Abbildung von $\mathbb{R}^3$ nach $\mathbb{R}$ als Observable o wählen. Dass o und auch die Lösungen von (2.1) in Abhängigkeit von der Zeit – also der mehrdimensionale Prozess selbst – allerdings eine gewisse Glattheitsanforderung erfüllen sollten und nicht gänzlich beliebig sind, werden wir in Kapitel 3 feststellen.

2.1.2 Begriffsklärung

Die zwei Ziele der Zeitreihenanalyse sind nun, anhand der gegebenen Zeitreihe

1. Aussagen über die Struktur des zugrundeliegenden Prozesses zu machen und

2. die zukünftige Entwicklung der Zeitreihe selbst vorherzusagen.

Will man allerdings die Güte der Vorhersage überprüfen, muss man davon ausgehen, dass es problemlos möglich ist, neue Zeitreihendaten zu erzeugen, die sich in ihrer weiteren Entwicklung ebenfalls exakt am vorher zugrundeliegenden Prozess orientieren. Dies würde in der Physik beispielsweise bedeuten, dass das Experiment zu den exakt gleichen Bedingungen wie zuvor weitergeführt werden kann. Die Hoffnung ist hierbei, dass sich der Prozess unter erneuter Beobachtung durch die Observable so verhält, wie er dies bei der ersten Beobachtung tat.
Leider ist dies nicht immer möglich, da z.B. gewisse Experimente zu teuer sind oder sich die Voraussetzungen geändert haben. Man sieht also, dass es in der Praxis oftmals nicht möglich ist, beliebig lange Zeitreihen zu erzeugen, bzw. bereits erzeugte Zeitreihen zu reproduzieren. Die zwangsläufig resultierende Endlichkeit der zu untersuchenden Zeitreihe ist bereits ein erstes Problem bei der praktischen Anwendung von Takens' Theorem. Nähere Erläuterungen hierzu werden in Kapitel 4 folgen.
Intuitiv ist auch klar, dass es bei einer "schlechten" Observable schwieriger oder sogar unmöglich wird, die Zeitreihe vorherzusagen und den Prozess aufgrund der Kenntnis der Zeitreihe genauer zu charakterisieren.
Würde man im obigen Beispiel 2.1

$$o\left((x_1, x_2, x_3)^T\right) \equiv 0$$

wählen, so wäre jedes Element der Zeitreihe $o\left((X(t_i), Y(t_i), Z(t_i))^T\right)$ gleich 0. Es ist offensichtlich, dass dies eine ungünstige Wahl einer Observablen o ist, da man zwar die Nullfolge perfekt vorhersagen kann, jedoch keinerlei Erkenntnisgewinn über den eigentlichen Prozess (2.1) hat. Was in der Praxis "gute" und "schlechte" Observablen sind, wird mit Hilfe von Takens' Theorem 3.1 deutlich werden.
Wir wollen nun eine möglichst allgemeine Definition einer Zeitreihe angeben, welche mit unserem bisherigen Verständnis und den Voraussetzungen in Takens' Theorem vereinbar ist.

Definition 2.1 [Zeitreihe (zeitdiskret)]
Sei M eine m-dimensionale C^2-Mannigfaltigkeit. Des Weiteren sei $\phi : M \to M$ ein C^2-Diffeomorphismus und $o : M \to \mathbb{R} \in C^2(M, \mathbb{R})$. $\mathbf{z_0} \in M$ sei ein beliebiger Punkt und weiter sei $N \in \mathbb{N} \cup \{\infty\}$. Dann nennen wir die Folge

$$\left(o\left(\phi^k(\mathbf{z}_0)\right)\right)_{k=0}^{N} \tag{2.2}$$

*eine **zeitdiskrete Zeitreihe**.*

Dies ist so zu interpretieren, dass der Verlauf unseres Prozesses auf der Mannigfaltigkeit M – die wir künftig auch als *Phasen-* oder *Zustandsraum* bezeichnen – durch die

Abbildung ϕ determiniert ist. Diese Definition ist zeitdiskret, da lediglich das Hintereinanderausführen von ϕ betrachtet wird.
Sei nun $\mathbb{R}^+ := [0, \infty)$. Eine zeitkontinuierliche Definition lautet wie folgt:

Definition 2.2 [Zeitreihe (zeitkontinuierlich)]
Seien M, o und $\mathbf{z_0}$ wie in Definition 2.1. Sei ferner $\mathbf{V} : M \to TM \in C^2$ ein Vektorfeld. $\mathbf{z} : \mathbb{R}^+ \to M$ erfülle folgende Differentialgleichung:

$$\frac{d\mathbf{z}}{dt} = \mathbf{V}(\mathbf{z}); \quad \mathbf{z}(0) = \mathbf{z_0}. \tag{2.3}$$

Dann ist

$$\phi_t(\mathbf{z_0}) = \mathbf{z}(t) \tag{2.4}$$

der Fluss zum Vektorfeld $\mathbf{V}$. Wir nennen

$$\left(o\left(\phi_t(\mathbf{z_0})\right)\right)_{t \in \mathbb{R}^+} \tag{2.5}$$

*eine **zeitkontinuierliche Zeitreihe**.*

Handelt es sich bei M nicht um den $\mathbb{R}^n$, so ist der gesamte Ausdruck in Karten der Mannigfaltigkeit zu verstehen. Die Differentiation von $\mathbf{z}$ nach t bezieht sich hierbei immer auf den *Levi-Civita-Zusammenhang* der Mannigfaltigkeit, weshalb wir korrekterweise fordern müssen, dass M zusätzlich *Riemannsch* ist. Wir verzichten aus Gründen der Überschaubarkeit bewusst auf die explizite Darstellung in Karten und wollen die implizite Zusatzvoraussetzung, dass M Riemannsch ist, hier nur aus Gründen der Vollständigkeit erwähnen. In der Praxis ist dies ohnehin nicht weiter von Belang, da M dort oftmals eine Teilmenge des $\mathbb{R}^n$ ist und wir die Identitätsfunktion als Karte wählen können. Hierdurch wird auch der Levi-Civita-Zusammenhang der Mannigfaltigkeit trivial und die Differentiation wird zur bekannten Richtungsableitung im $\mathbb{R}^n$. Für genauere Betrachtungen der hier erwähnten Konzepte verweisen wir erneut auf [doC92].
Die C^2-Glattheitsbedingungen in den Definitionen 2.1 und 2.2 sind an die Glattheitsbedingungen in Takens' Theorem angepasst. Es würde hier durchaus sinnvoll sein, auch bei nicht-glatten Funktionen oder Mannigfaltigkeiten in der Definition von "Zeitreihen" zu sprechen. Bei unseren Experimenten in Kapitel 7 werden wir den Begriff der Zeitreihe wesentlich intuitiver verwenden und beliebige zeitlich indizierte Folgen $(a_t)_t$ – unabhängig von eventuell zugrundeliegenden glatten mehrdimensionalen Prozessen – als Zeitreihe bezeichnen.
Legen wir die Flussformulierung (2.4) einer Zeitreihe zugrunde, so wird die Entwicklung des Prozesses durch ϕ_t und $\mathbf{z_0}$ bestimmt. Wie bereits erwähnt, handelt es sich bei (2.3) um ein System von gewöhnlichen Differentialgleichungen erster Ordnung. Daher wissen wir, dass $\mathbf{z}(t)$ – und somit auch $\phi_t(\mathbf{z_0})$ – eindeutig durch $\mathbf{V}$ und $\mathbf{z_0}$ bestimmt wird. Dadurch geraten wir in die im Abschnitt 2.1 beschriebene Situation, dass der gesamte Verlauf – also die *Trajektorie* – des zukünftigen Prozesses für ein festes Vektorfeld $\mathbf{V}$ durch den Anfangspunkt $\mathbf{z_0}$ determiniert ist. Wir erhalten somit die gewünschte 1-zu-1

Beziehung zwischen aktuellem Zustand – kodiert als Punkt in M – und der künftigen Trajektorie:

$$\mathbf{z_0} \quad \overset{1:1}{\longleftrightarrow} \quad \phi_t(\mathbf{z_0}) \; \forall \; t \in \mathbb{R}^+. \tag{2.6}$$

Diese Beziehung ist im obigen Sinne zu verstehen: Für ein festes $\mathbf{z_0}$ ist $\phi_t(\mathbf{z_0})$ vollständig bestimmt. Trivialerweise gilt dies auch umgekehrt.
Dass eine analoge 1-zu-1-Beziehung auch im diskreten Fall vorliegt, ist sofort ersichtlich, da der Wert jeder Iterierten $\phi^k(\mathbf{z_0})$ lediglich von $\mathbf{z_0}$ abhängt.

2.1.3 Erneute Betrachtung des Lorenz-Systems

Um die vorgestellten Konzepte zu verdeutlichen, greifen wir das Beispiel 2.1 erneut auf:

Beispiel 2.2: [Lorenz-System (Begriffsklärung)] Wir betrachten das bereits bekannte Lorenz-System aus Gleichung (2.1):

$$\begin{aligned} \dot{X} &= a(Y - X) \\ \dot{Y} &= X(b - Z) - Y \\ \dot{Z} &= XY - cZ \end{aligned}$$

Wie wir in Unterabschnitt 2.1.1 gesehen haben, führt dieses System zu einer kontinuierlichen Zeitreihe. Wir wollen es also im Kontext von Definition 2.2 betrachten.

- Der Vektorraum $\mathbb{R}^3$ stellt hier die C^∞-Mannigfaltigkeit M dar.

- $\mathbf{z_0} \in \mathbb{R}^3$ ist ein beliebiger Punkt, der die Anfangsbedingungen der Differentialgleichung darstellt.

- Die Observable $o \in C^2(\mathbb{R}^3, \mathbb{R})$ kann beliebig gewählt werden.

- Für das Vektorfeld $\mathbf{V}$ gilt

  $$\mathbf{V} : \mathbb{R}^3 \to T(\mathbb{R}^3) \cong \mathbb{R}^3,$$

  da wir den Tangentialraum des $\mathbb{R}^n$ immer mit dem $\mathbb{R}^n$ selbst identifizieren können. V hat also folgende Gestalt:

  $$\mathbf{V}\left(\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}\right) = \begin{pmatrix} a \cdot (x_2 - x_1) \\ x_1 \cdot (b - x_3) - x_2 \\ x_1 \cdot x_2 - c \cdot x_3 \end{pmatrix}.$$

- $\mathbf{z} : \mathbb{R}^3 \to \mathbb{R}^3$ muss also folgendes Differentialgleichungssystem erfüllen:

$$\left[\begin{array}{rcl} \frac{\partial z_1}{\partial t}(t) & = & a \cdot (z_2(t) - z_1(t)) \\ \frac{\partial z_2}{\partial t}(t) & = & z_1(t) \cdot (b - z_3(t)) - z_2(t) \\ \frac{\partial z_3}{\partial t}(t) & = & z_1(t) \cdot z_2(t) - c \cdot z_3(t) \end{array}\right], \quad \mathbf{z}(0) = \mathbf{z_0}.$$

z_1, z_2 und z_3 sind hierbei die Komponenten von $\mathbf{z}$: $\mathbf{z}(t) = (z_1(t), z_2(t), z_3(t))^T$.

Unsere resultierende Zeitreihe ist damit

$$\left(o\left(\mathbf{z}(t)\right)\right)_{t \in \mathbb{R}^+}.$$

2.2 Die Entwicklung der Trajektorie

Wir werden nun Bereiche diskutieren, die normalerweise eine fundierte chaostheoretische Grundlage benötigen. Unser Hauptaugenmerk soll hier aber darauf liegen, die für diese Diplomarbeit wichtigen Konzepte möglichst verständlich zu erklären, ohne dabei auf das chaostheoretische Fundament (z.B. *Lyapunov-Exponenten*) genauer einzugehen. Der Begriff "Chaos" ist hier nicht direkt mit seiner umgangssprachlichen Bedeutung von "Durcheinander" zu identifizieren. Wir wollen Chaos wie folgt verstehen: Die Entwicklung der Trajektorie eines Systems ist nicht stabil hinsichtlich der Anfangsbedingungen. Anschaulich bedeutet dies, dass zwei minimal unterschiedliche Anfangsbedingungen eines chaotischen Prozesses nach einiger Zeit zu zwei völlig verschiedenen Trajektorien führen können. Da wir die Anfangsbedingungen eines physikalischen Prozesses meistens nur bis zu einer gewissen Genauigkeit kennen, und unsere Beobachtungen ohnehin einem Messfehler unterliegen, kann dies in der Praxis zu gravierend falschen Ergebnissen führen. Zur Einführung in die relevanten chaostheoretischen Konzepte sei auf [Rue91, KS04] verwiesen.

2.2.1 Dissipative Prozesse

Meist ist zu beobachten, dass Trajektorien

- entweder gegen unendlich streben

$$||\phi_t(\mathbf{z_0})||_2 \quad \stackrel{t \to \infty}{\longrightarrow} \quad \infty \tag{2.7}$$

- oder für immer in einem begrenzten Gebiet verlaufen

$$\overline{\lim_{t \to \infty}} ||\phi_t(\mathbf{z_0})||_2 \leq c < \infty. \tag{2.8}$$

Dies hängt von der Wahl der Anfangsbedingung $\mathbf{z_0}$ und zusätzlichen Parametern der Differentialgleichung ab. Wir interessieren uns hier lediglich für den zweiten Fall (2.8), da wir sonst keine Möglichkeit haben, Takens' Theorem anzuwenden, wie wir in Abschnitt 3.1 sehen werden.
In vielen relevanten Fällen ist es nicht nur so, dass die Trajektorie $\phi_t(\mathbf{z_0})$ beschränkt ist, sondern es liegt zusätzlich auch eine Art *Raumkontraktion* vor:

Definition 2.3 [Dissipation]
Man nennt den durch einen Diffeomorphismus $\phi : M \to M$ wie in Definition 2.1 definierten ***zeitdiskreten Prozess***

$$\mathbf{z_{i+1}} = \phi\left(\mathbf{z_i}\right); \quad i \in \mathbb{N} \tag{2.9}$$

auf $N \subseteq M$ ***dissipativ****, falls*

$$|\det \mathbf{J}_\phi(\mathbf{x})| < 1 \tag{2.10}$$

für alle $\mathbf{x} \in N$ gilt. Hierbei ist $\mathbf{J}_\phi$ die Jacobi-Matrix von ϕ:

$$(\mathbf{J}_\phi)_{ij} = \frac{\partial}{\partial x_j}\phi_i(\mathbf{x}).$$

Des Weiteren nennt man den durch ein Vektorfeld $\mathbf{X} : M \to TM$ wie in Definition 2.2 definierten ***zeitkontinuierlichen Prozess*** *ϕ_t auf $N \subseteq M$* ***dissipativ****, falls*

$$\operatorname{div} \mathbf{X}(\mathbf{x}) < 0 \tag{2.11}$$

für alle $\mathbf{x} \in N$ gilt.

Für Trajektorien mechanischer Prozesse ist Dissipation gleichbedeutend mit dem Verlust kinetischer Energie. Diese wird beispielsweise in Wärmeenergie umgewandelt. Im Allgemeinen kann man Dissipation so verstehen, dass das System nur dann nicht zum Stillstand kommt, wenn von außen Energie zugeführt wird. Man sieht also, dass die Voraussetzung der Dissipation keine strenge Forderung an den zu beobachtenden Prozess darstellt, sondern oftmals eine durchaus realistische Eigenschaft eines Systems ist.
Dass es berechtigt ist, bei dissipativen Prozessen von einer Raumkontraktion zu sprechen, ist schnell einzusehen:
Für dissipative Prozesse gilt, dass sie das Volumen einer Nicht-Nullmenge von Anfangsbedinungen verkleinern. Hierbei orientieren wir uns am Satz von Liouville [Kus04]. Für das Volumen V einer Menge D (mit stückweise glattem Rand ∂D) von zulässigen Anfangsbedingungen im Phasenraum gilt:

$$\frac{\partial V}{\partial t} = \int_D \operatorname{div} \mathbf{X}(\mathbf{x})\, d\mathbf{x}. \tag{2.12}$$

Wie man leicht sehen kann, repräsentiert diese Gleichung lediglich den Energieerhaltungssatz im Sinne des Satzes von Gauß [Gri06b]:

$$\oint_{\partial D} \mathbf{X}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) \, d\mathbf{x} \quad = \quad \int_D \operatorname{div} \mathbf{X}(\mathbf{x}) \, d\mathbf{x}. \tag{2.13}$$

- $\mathbf{n}(\mathbf{x})$ ist hierbei die nach außen gerichtete Normale am Punkt $\mathbf{x}$ der Oberfläche ∂D.
- Der **linke Term** ist eine Bilanz des durch $\mathbf{X}$ verursachten Ein- und Ausfluss an der Oberfläche.
- Der **rechte Term** repräsentiert alle Quellen und Senken innerhalb der Menge D.

Für negative Divergenz ist nun auch die linke Seite von (2.13) negativ und somit gibt es mehr Fluss in das Volumen hinein als heraus. Dies bedeutet, dass die Trajektorien in das Volumen "hineingezogen" werden.
Für einen diskreten dissipativen Prozess gilt analog: Ist D wie oben und ϕ wie in Gleichung (2.9), so ist

$$\operatorname{vol}(\phi(D)) \; = \; \int_{\phi(D)} d\mathbf{x} \; \overset{(*)}{=} \; \int_D |\det \mathbf{J}_\phi(\mathbf{x})| \; d\mathbf{x} \; \overset{(2.10)}{<} \; \int_D d\mathbf{x} \; = \; \operatorname{vol}(D). \tag{2.14}$$

Bei $(*)$ haben wir den Transformationssatz verwendet.
Bei dissipativen Prozessen liegt eine noch stärkere Eigenschaft vor. Man stellt hier die Kontraktion auf eine Nullmenge fest:

Satz 2.4 [Kontraktion auf eine Nullmenge]
Für einen dissipativen Prozess gilt: Für eine beschränkte Menge von zulässigen Anfangsbedingungen D_0 mit Volumen $\operatorname{vol}(D_0)$ *ist*

$$\lim_{t \to \infty} \operatorname{vol}(D_t) = 0. \tag{2.15}$$

Hierbei stellt D_t das Bild von D_0 zum Zeitpunkt t dar.

Beweis. Wir betrachten den Abschluss $D = \overline{D_0}$, der dasselbe Volumen wie D_0 besitzt und alle Punkte enthält, die auch in D_0 enthalten sind. Gilt der Satz für D, so gilt er auch für D_0.

- Im **zeitdiskreten Fall** liegt ein C^2-Diffeomorphismus $\phi : M \to M$ wie in Definition 2.1 vor. Somit ist $\det \mathbf{J}_\phi$ stetig und nimmt auf D Maximum und Minimum an. Dies bedeutet, dass auf D

  $$|\det \mathbf{J}_\phi| \leq c_1 < 1,$$

  gilt.

Analog zu Gleichung (2.14) gilt:

$$\mathrm{vol}(\phi(D)) \leq c_1 \cdot \mathrm{vol}(D).$$

Somit erhalten wir:

$$\mathrm{vol}(D_n) \leq c_1^n \cdot \mathrm{vol}(D_0) \quad \overset{n\to\infty}{\to} 0.$$

- Im **zeitkontinuierlichen Fall** liegt ein C^2-Vektorfeld $\mathbf{X} : M \to TM$ wie in Definition 2.2 vor, für welches $\mathrm{div}\,\mathbf{X}$ stetig ist und auf D Maximum und Minimum annimmt. Somit gilt

  $$\mathrm{div}\,\mathbf{X} \leq c_2 < 0$$

  auf D. Wir orientieren uns an Gleichung (2.12) und erhalten

  $$\frac{\partial}{\partial t}\mathrm{vol}(D_t) \leq c_2 \cdot \mathrm{vol}(D_t).$$

  Damit gilt

  $$\mathrm{vol}(D_t) \leq \mathrm{vol}(D_0) \cdot e^{c_2 \cdot t} \quad \overset{t\to\infty}{\to} 0.$$

□

2.2.2 Attraktoren

Der beschriebene Kontraktionseffekt führt nun dazu, dass sich verschiedene Trajektorien mit Anfangsbedingungen aus der Menge D auf eine Nullmenge A zubewegen. In der Praxis gelten oftmals folgende Eigenschaften, die jedoch nicht einfach zu verifizieren sind:

- A selbst bleibt invariant unter der Entwicklung des Prozesses:

  $$\phi_t(\mathbf{z_0}) \in A \;\; \forall\, \mathbf{z_0} \in A, t \in \mathbb{R}^+ \tag{2.16}$$

- Es gibt eine Nachbarschaft $V \supsetneq A$, sodass:

  $$\begin{array}{ll} & \forall\, \mathbf{p} \in V, \delta \in \mathbb{R}^{++}: \\ \textbf{(1)} & \phi_t(\mathbf{p}) \in V \;\; \forall\, t \in \mathbb{R}^+ \\ \textbf{(2)} & \exists\, T \in \mathbb{R}^+ : \;\; \phi_t(\mathbf{p}) \in U_\delta(A) \;\; \forall\, t > T. \end{array} \tag{2.17}$$

  $\mathbb{R}^{++}$ bezeichnet hier das reelle Intervall $(0, \infty)$. $U_\delta(A)$ ist die δ-Umgebung von A:

  $$U_\delta(A) = \{\mathbf{x} \in M |\; d(\mathbf{x}, A) < \delta\}.$$

  Hierbei ist d das durch die Riemannsche Metrik definierte Abstandsmaß und

  $$d(\mathbf{x}, A) = \inf_{\mathbf{y}\in A} d(\mathbf{x}, \mathbf{y}).$$

Den degenerierten Fall $V = A$ wollen wir nicht zulassen.

Es sei darauf hingewiesen, dass wir für Prozesse wie in Definition 2.3 jede kompakte Menge von gültigen Anfangsbedingungen, die A enthält, als Nachbarschaft V wählen können, siehe [Gru04]. Es gibt aber durchaus Prozesse, die nur in einem Teil des Raumes dissipativ verlaufen. Für diese wird die mögliche Nachbarschaft V entsprechend kleiner und es kann mehrere Mengen mit den Eigenschaften von A geben, die voneinader disjunkt sind.

- Für A gilt:
$$\nexists\, B \subsetneq A : \ B \text{ erfüllt (2.16) und (2.17).} \tag{2.18}$$

Definition 2.5 [Attraktor]
Eine Menge A, die (2.16), (2.17) *und* (2.18) *erfüllt, nennen wir* ***Attraktor****.*

Die Gestalt eines Attraktors ist in der Praxis häufig so komplex, dass dieser keine Mannigfaltigkeit mehr ist. Näheres hierzu folgt in Kapitel 4.
Oftmals ist es schwierig, korrekte und exakte Beschreibungen für Attraktoren zu finden. Teilweise ist nicht einmal deren Existenz geklärt. Allerdings gibt es für dissipative Prozesse, wie in Definition 2.3, immer einen globalen Attraktor, d.h. jede beschränkte Menge kann als V in Gleichung (2.17) gewählt werden. Oftmals kann für Prozesse, die nur auf einem Teil des Raums dissipativ sind, unter wenigen Zusatzannahmen die Existenz eines lokalen Attraktors nachgewiesen werden. Eine solche Zusatzannahme ist z.B.

$$\exists\, W \subset M, t_0 \in \mathbb{R}^+ : \phi_t \text{ ist dissipativ auf } W \text{ und } \phi_t(W) \subset W \ \forall t > t_0.$$

Für Beweise und genauere Abhandlungen dieses Themas sei auf [Gru04] verwiesen. Genauere Betrachtungen zum Zusammenhang der Gestalt eines Prozesses und der eines Attraktors findet man in [KS04]. Die Existenz von Attraktoren werden wir hier nicht explizit nachprüfen, sondern lediglich a posteriori – anhand von Trajektorien der untersuchten Prozesse – vermuten können.

2.2.3 Weitere Beispiele für dynamische Systeme

Wir wollen nun neben dem bereits aus den Beispielen 2.1 und 2.2 bekannten Lorenz-System auch andere Prozesse untersuchen, um die oben eingeführten Begriffe zu verdeutlichen.

Beispiel 2.3: [Konservatives System] Unter *konservativen Systemen* verstehen wir Systeme, die statt der Dissipationsbedingung aus Definition 2.3 folgende Bedingung erfüllen:

- $|\det \mathbf{J}_\phi| = 1$ **(Diskreter Prozess)**, bzw.
- $\operatorname{div} \mathbf{X} = 0$ **(Kontinuierlicher Prozess)**.

Wie man leicht (analog zu den Gleichungen (2.12) und (2.14)) sehen kann, sind dies Prozesse, die Volumen erhalten, statt zu kontrahieren. Ein einfaches kontinuierliches Beispiel im $\mathbb{R}^2$ finden wir im folgenden Differentialgleichungssystem, welches aus [KS04] entnommen wurde:

$$\begin{array}{c} \dot{x} = -\omega y \\ \dot{y} = \omega x \end{array} , \quad \omega \in \mathbb{R}. \tag{2.19}$$

Die Divergenz des korrespondierenden Vektorfeldes $\mathbf{X}$ ergibt sich als

$$\operatorname{div} \mathbf{X} = -\underbrace{\frac{\partial}{\partial x}\omega y}_{=0} + \underbrace{\frac{\partial}{\partial y}\omega x}_{=0} = 0.$$

Die Volumenerhaltung wird noch deutlicher, wenn man die Lösungen zu (2.19) betrachtet. Diese sind von der Form:

$$\left(\begin{array}{c} x(t) = x_0 \cdot \cos(\omega t) + y_0 \cdot \sin(\omega t) \\ y(t) = x_0 \cdot \sin(\omega t) + y_0 \cdot \cos(\omega t) \end{array} \right), \quad a \in \mathbb{R}, \quad \left(\begin{array}{c} x(0) = x_0 \\ y(0) = y_0 \end{array} \right) \in \mathbb{R}^2 \text{ beliebig.}$$

Das Maß einer beliebigen Menge von Anfangsbedingungen im $\mathbb{R}^2$ mit stückweise glattem Rand bleibt hier erhalten, da jede Trajektorie des Prozesses periodisch ist und jeder Anfangspunkt immer wieder besucht wird. Die Menge der Punkte einer Trajektorie bildet hier – nach Definition 2.5 – **keinen** Attraktor. Die Voraussetzung (2.16) ist aufgrund der Periodizität der Trajektorien zwar erfüllt, (2.17) gilt jedoch nicht, da $V = A$ die größtmögliche Nachbarschaft wäre, auf der diese Eigenschaft gilt. Von einer "Attraktion" der näheren Umgebung kann hier also keine Rede sein, da jede Trajektorie ein periodisches, invariantes System bildet.

Beispiel 2.4: [Lorenz-System (Dissipation und Attraktor)] Wir wollen nun das Lorenz-System aus Gleichung (2.1) auf Dissipation und Attraktoren untersuchen. Erneut betrachten wir das zugehörige Vektorfeld $\mathbf{X}$, welches wir bereits in Beispiel 2.2 gesehen haben:

$$\mathbf{X}\left(\left(\begin{array}{c} x_1 \\ x_2 \\ x_3 \end{array} \right) \right) = \left(\begin{array}{c} a \cdot (x_2 - x_1) \\ x_1 \cdot (b - x_3) - x_2 \\ x_1 \cdot x_2 - c \cdot x_3 \end{array} \right).$$

Die Divergenz von $\mathbf{X}$ ist nun

$$\begin{aligned} \operatorname{div} \mathbf{X} &= \frac{\partial}{\partial x_1}(a \cdot (x_2 - x_1)) + \frac{\partial}{\partial x_2}(x_1 \cdot (b - x_3) - x_2) + \frac{\partial}{\partial x_3}(x_1 \cdot x_2 - c \cdot x_3) \\ &= -(a + 1 + c). \end{aligned}$$

Somit ist der Prozess für Parameter aus

$$\{ \ (a, b, c)^T \in \mathbb{R}^3 \ \mid a + c > -1 \ \}$$

dissipativ und kontrahiert das Phasenraumvolumen. Da die Divergenz von V nicht von den Phasenraumkoordinaten x_1, x_2, x_3 abhängt, ist der Prozess für geeignete Parameter a und c auf dem gesamten Raum dissipativ und es existiert somit auch ein (globaler) Attraktor. In realistischen Fällen ist $a + c$ ungefähr 9 – siehe [KS04] – wodurch die Dissipationsbedingung erfüllt wäre.
In Abbildung 2.1 sind verschiedene Trajektorien dargestellt.

Beispiel 2.5: [Henon-Abbildung] Zuletzt wollen wir noch ein Beispiel für einen zeitdiskreten Prozess betrachen: Die sogenannte *Henon*-Abbildung [Hen76] ist durch

$$h_{n+1} = a - h_n^2 + b \cdot h_{n-1} \ , \quad a, b \in \mathbb{R} \tag{2.20}$$

für reelle $h_i, i \in \mathbb{N}$ definiert. Die Anfangsbedingungen $h_0, h_1 \in \mathbb{R}$ sind frei wählbar.
Wir wollen dies mit den von uns eingeführten Begriffen vereinbaren und definieren durch

$$\phi\left(\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}\right) = \begin{pmatrix} a - x_1^2 + b \cdot x_2 \\ x_1 \end{pmatrix} \tag{2.21}$$

einen Diffeomorphismus $\phi : \mathbb{R}^2 \to \mathbb{R}^2$. Nun befinden wir uns wieder im Kontext von Definition 2.1. Die Henon-Abbildung kann man nun bereits selbst als Zeitreihe auffassen. Die zu verwendende Beobachtungsfunktion ist

$$o\left(\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}\right) = x_2.$$

Dann gilt nach der obigen Definition der Henon-Abbildung:

$$o\left(\phi^k \begin{pmatrix} h_1 \\ h_0 \end{pmatrix}\right) = o\left(\phi^{k-1} \begin{pmatrix} \overbrace{a - h_1^2 + b \cdot h_0}^{=h_2} \\ h_1 \end{pmatrix}\right) = \ldots = o\left(\begin{pmatrix} h_{k+1} \\ h_k \end{pmatrix}\right) = h_k. \tag{2.22}$$

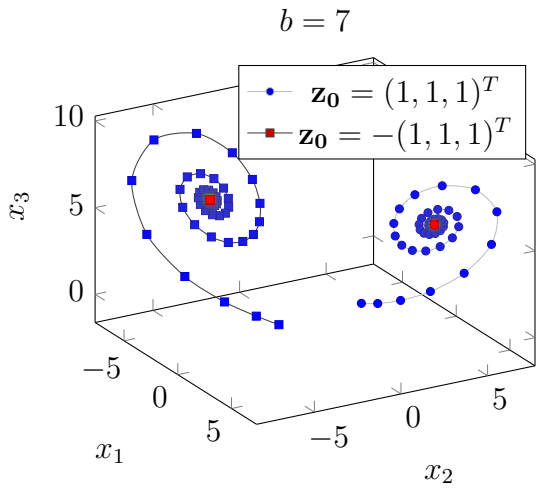
Für ϕ gilt nun

$$\mathbf{J}_\phi = \begin{pmatrix} \frac{\partial}{\partial x_1}(a - x_1^2 + b \cdot x_2) & \frac{\partial}{\partial x_2}(a - x_1^2 + b \cdot x_2) \\ \frac{\partial}{\partial x_1} x_1 & \frac{\partial}{\partial x_2} x_1 \end{pmatrix} = \begin{pmatrix} 2 \cdot x_1 & b \\ 1 & 0 \end{pmatrix}$$

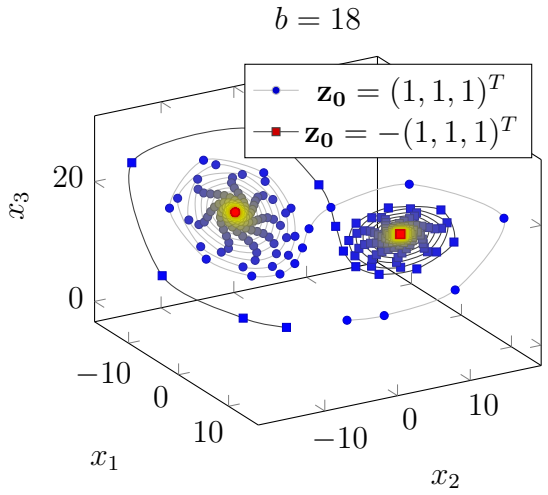
und somit

$$|\det \mathbf{J}_\phi| \equiv |b|.$$

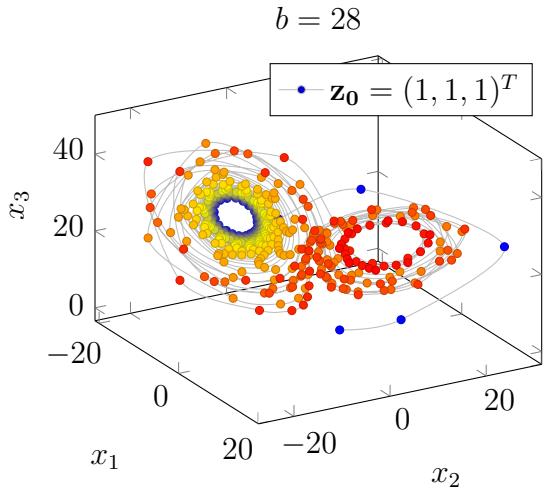
Somit ist der Prozess für $-1 < b < 1$ dissipativ und für $b = \pm 1$ konservativ. Da auch hier die Dissipation nur von b und nicht von den Koordinaten im Phasenraum abhängt, handelt es sich für $|b| < 1$ um einen im gesamten Phasenraum dissipativen Prozess. Somit liegt ein (globaler) Attraktor vor.
In Abbildung 2.2 sind verschiedene Trajektorien dargestellt.



(a) Punktattraktoren

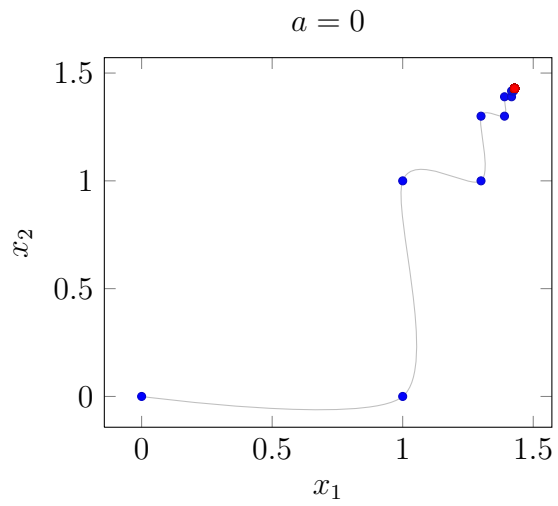


(b) Punktattraktoren – längere Transienz-Phase als bei (a)

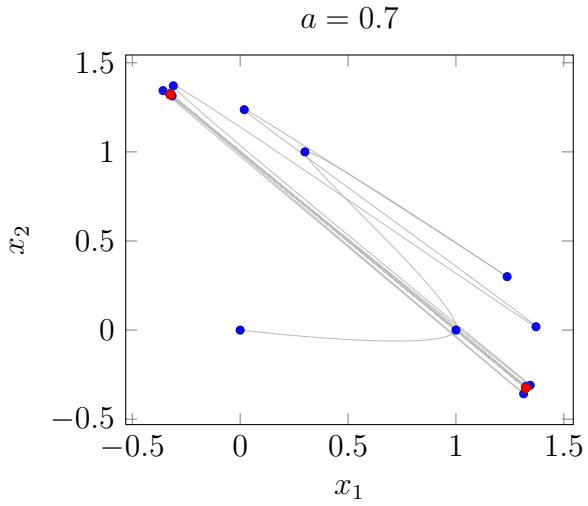


(c) Chaotischer Attraktor

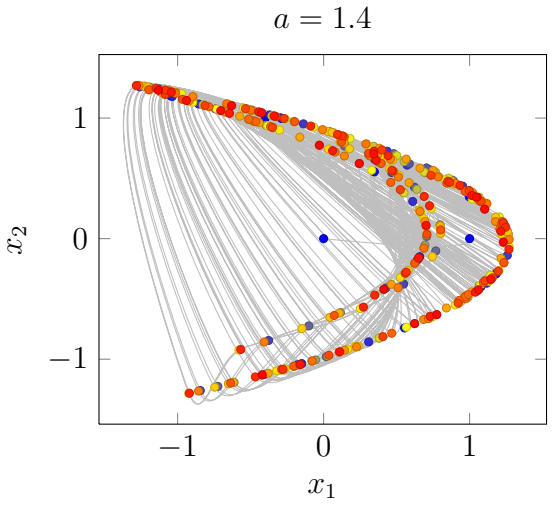
Abb. 2.1: Zu sehen sind verschiedene Trajektorien des Lorenz-Systems für $a = 10$, $c = \frac{8}{3}$ und unterschiedliche b, welche mittels einem Runge-Kutta-Verfahren vierter Ordnung (Schrittweite 0.1) berechnet wurden. Dargestellt sind jeweils die ersten 400 Punkte der Trajektorien. Die Zeitskala ist hierbei durch den Farbverlauf von blau nach rot dargestellt. In (a) und (b) bewegen sich die Trajektorien auf einen einzelnen Punkt zu. In (c) sieht man eine Bewegung auf einem *chaotischen Attraktor*. Einen Attraktor nennt man chaotisch, falls dieser ein Fraktal ist und dessen Dimension somit nicht ganzzahlig ist. Für eine genauere Erläuterung des Dimensionsbegriffs sei auf Abschnitt 4.2 verwiesen.



(a) 0-dimensionaler Attraktor – Ein Punkt



(b) 0-dimensionaler Attraktor – Zwei Punkte



(c) Chaotischer Attraktor

Abb. 2.2: Zu sehen sind verschiedene Trajektorien des Henon-Systems für $b = 0.3$, unterschiedliche a und $\mathbf{z_0} = (0, 0)^T$. Dargestellt sind jeweils die ersten 400 Punkte der Trajektorien. Die Zeitskala wird hierbei mittels des Farbverlaufs von blau nach rot dargestellt. In (a) bewegt sich die Trajektorie auf einen einzelnen Punkt zu. In (b) springt die Trajektorie nach einer Phase der Transienz zwischen zwei Punkten. In (c) sieht man eine Bewegung auf einem chaotischen Attraktor.

3 Takens' Theorem

Auch wenn die Wahl des Kapitelnames etwas anderes suggeriert, gibt es mehrere wichtige, von Floris Takens stammende Theoreme, die für diese Arbeit grundlegend sind. Meistens spricht man jedoch von *Takens' Theorem* und bezieht sich damit auf Theorem 1 aus [Tak81], welches wir hier ebenfalls unter diesem Namen einführen werden. Interessant wird im Folgenden vor allem sein, dass die Voraussetzungen für die Anwendung von Takens' Theorem sehr allgemein sind. Dies ist letztendlich der Grund, warum die folgenden Sätze auch in praktischen Anwendungen relevant sind. Allerdings werden wir in Kapitel 4 sehen, dass wir bei der Analyse von Zeitreihen auf neue Probleme treffen, die in der gesamten Theorie um Takens' Theoreme nur eine untergeordnete oder gar keine Rolle spielen.

3.1 Einbettung nach Takens

Bisher haben wir uns fast ausschließlich auf den Prozess im Phasenraum konzentriert. Nun wenden wir uns der vorliegenden Zeitreihe zu. Die Hoffnung ist, dass die Kenntnis der Zeitreihe Rückschlüsse auf den zugrundeliegenden Prozess erlaubt. Um dies nochmal hervorzuheben: Sind die Anfangsbedingung und der Prozess im Phasenraum bekannt, so ist die Trajektorie vollständig determiniert und kann fehlerfrei vorhergesagt werden. Wir wollen nun also aufgrund der Zeitreihe in $\mathbb{R}$ den Prozess in M rekonstruieren. Dies ist nicht direkt möglich, allerdings liefert Takens' Theorem einen guten Kompromiss. Die etwas abstrakte Definition 2.1 einer diskreten Zeitreihe passt nun perfekt in den Kontext von Takens' Theorem:

Theorem 3.1 [Takens' Theorem (zeitdiskrete Delay-Einbettung)]
Sei M_0 eine kompakte m-dimensionale C^2-Mannigfaltigkeit, $\phi : M_0 \to M_0$ ein C^2-Diffeomorphismus und $o : M_0 \to \mathbb{R} \in C^2(M_0, \mathbb{R})$. Dann ist die Abbildung $\mathbf{\Phi}_{(\phi,o)} : M_0 \to \mathbb{R}^{2m+1}$, welche durch

$$\mathbf{\Phi}_{(\phi,o)}(\mathbf{x}) := \big(o(\mathbf{x}), o(\phi(\mathbf{x})), o(\phi^2(\mathbf{x})), \ldots, o(\phi^{2m}(\mathbf{x}))\big) \tag{3.1}$$

*definiert ist, **im Allgemeinen** eine Einbettung.*

Der Begriff der Einbettung ist hier als Diffeomorphismus auf das Bild von $\mathbf{\Phi}$ zu verstehen. Dies bedeutet also, dass wir die Möglichkeit haben, anhand der Zeitreihe eine Untermannigfaltigkeit – also eine offene Menge – im $\mathbb{R}^{2m+1}$ zu konstruieren, die sich diffeomorph zum ursprünglichen Phasenraum verhält. Dies ermöglicht uns, die Trajektorie im $\mathbb{R}^{2m+1}$ zu untersuchen. Auch wenn Dissipation nicht zwangsläufig erhalten bleibt, da z.B. im

diskreten Fall

$$|\det \mathbf{J}_{\mathbf{\Phi}\circ\phi}| = |\det \mathbf{J}_{\mathbf{\Phi}}| \cdot |\det \mathbf{J}_{\phi}|$$

gilt und $\det \mathbf{J}_{\mathbf{\Phi}}$ beliebig groß werden kann, so sind Bilder von Attraktoren trotzdem wieder Attraktoren für den Prozess im $\mathbb{R}^{2m+1}$ [Eng91]. Ohne bereits auf den Dimensionsbegriff näher eingehen zu wollen, sei angemerkt, dass sich auch die Attraktordimensionen nicht verändern.
Um dies zu verdeutlichen: Wenn Takens' Theorem anwendbar ist, erhalten wir einen Prozess im $\mathbb{R}^{2m+1}$, der diffeomorph zum ursprünglichen Prozess ist. Ist die Trajektorie eines Prozesses bereits durch die Anfangsbedingung determiniert, so ist es das diffeomorphe Gegenstück ebenfalls. Ist also ein $\mathbf{z_0} \in M_0$ als Anfangsbedingung vorgegeben, so determinieren wir die gesamte Trajektorie im Phasenraum. Umgekehrt gilt aber, dass wir aufgrund der Kenntnis von $\mathbf{\Phi}(\mathbf{z_0})$ auch $\mathbf{\Phi}(\phi^n(\mathbf{z_0})), n \in \mathbb{N}$ determiniert haben:
Da

$$\mathbf{\Phi}(\mathbf{z_0}) = \big(o(\mathbf{z_0}), o(\phi(\mathbf{z_0})), o(\phi^2(\mathbf{z_0})), \ldots, o(\phi^{2m}(\mathbf{z_0}))\big)$$

ist, legen wir den weiteren Prozess im $\mathbb{R}^{2m+1}$ und somit

$$\mathbf{\Phi}(\phi^n(\mathbf{z_0})) = \big(o(\phi^n(\mathbf{z_0})), o(\phi^{n+1}(\mathbf{z_0})), o(\phi^{n+2}(\mathbf{z_0})), \ldots, o(\phi^{n+2m}(\mathbf{z_0}))\big)$$

fest. Das heißt aber, dass wir durch die Kenntnis über den diffeomorphen Prozess in $\mathbb{R}^{2m+1}$ auch die Zeitreihe selbst determiniert haben. Diese eigentlich triviale Folgerung ist besonders hervorzuheben, weil sie es möglich macht, Lernalgorithmen im $\mathbb{R}^{2m+1}$ zu verwenden, um die Zeitreihe vorherzusagen. Dazu kommen wir in Kapitel 5.

3.1.1 Kompaktheit der Mannigfaltigkeit M_0

Interessant und keineswegs zu vernachlässigen ist die Voraussetzung der Kompaktheit von M_0. Wichtig ist hierbei, den Begriff nicht mit der Kompaktheit von Mengen zu verwechseln. Eine wichtige Eigenschaft einer kompakten Mannigfaltigkeit ist Geschlossenheit – nicht zu verwechseln mit Abgeschlossenheit. Zum Nachschlagen der Begriffe wird [doC92] empfohlen.
Die Kompaktheit macht es unmöglich, unter M_0 immer dasselbe zu verstehen wie unter dem Phasenraum M in Kapitel 2, da wir dort oftmals $M = \mathbb{R}^n$ gesetzt hatten. Wir benötigen eigentlich eine kompakte Untermannigfaltigkeit M_0, die unseren Attraktor enthält. Es gilt also

$$A \subseteq M_0 \subseteq M.$$

Im Grunde genommen ist dies eine Forderung an den Attraktor A selbst. Dieser muss eine Gestalt haben, die einfach genug ist, um sie in eine kompakte Untermannigfaltigkeit von M zu legen. Dies ist meist nicht einfach zu verifizieren, doch mit der Arbeit von Sauer, Yorke und Casdagli [SYC91] wurde es möglich, diese Voraussetzung fallen zu lassen. Dort wurde gezeigt, dass die 1-zu-1 Beziehung der Einbettung auch für beliebige kompakte Mengen A gilt und die Abbildung $\mathbf{\Phi}$ eine Immersion auf jede kompakte Un-

termenge einer Mannigfaltigkeit $\hat{M}$ ist, die in A enthalten ist. Nun benötigen wir also keine kompakte Mannigfaltigkeit mehr, die den Attraktor beinhaltet. Die zu verwendende Dimension m wird hierbei als die *Boxcounting-Dimension* von A gewählt. Die Einbettung findet schließlich im $\mathbb{R}^{\lceil 2m+1 \rceil}$ statt. Die Boxcounting-Dimension wird in Kapitel 4 erläutert.
Die Tatsache, dass die Norm des Prozesses nicht beliebig groß werden darf und wir uns daher nur im Kontext von Gleichung (2.8) bewegen können ist unabhängig davon, ob man die Formulierung von Sauer oder die von Takens zugrunde legt. Da Trajektorien wie in Gleichung (2.7) sich weder auf Attraktoren zubewegen, noch im Kontext von Takens' Theorem verwendet werden können, sind diese für uns nicht von Belang. Da wir aber in Abschnitt 2.2 gesehen haben, dass relevante realistische Prozesse zumindest auf einem Teil des Raumes dissipativ sind, fällt der Fall aus (2.7) ohnehin nicht in unser Interessengebiet.

3.1.2 Einschränkungen an ϕ und o

Weiterhin tritt in Takens' Theorem der Begriff "im Allgemeinen" auf. Dies soll das Folgende bedeuten:

> *Jede messbare Teilmenge von Kombinationen aus Observablen $o : M_0 \to \mathbb{R}$ und Diffeomorphismen $\phi : M \to M$, für die $\mathbf{\Phi}_{(\phi,o)}$ keine Einbettung ist, bildet eine Nullmenge bezüglich des Produktmaßes in $C^2(M_0, \mathbb{R}) \otimes Diff^2(M_0, M_0)$.*

Somit ist die Wahrscheinlichkeit, dass in der Praxis "versehentlich" solche o und ϕ vorliegen gleich 0. Liegen wir jedoch z.B. mit $\hat{o}$ – in der durch die Riemannsche Metrik induzierten l^∞-Operatornorm – nahe bei einem solchen o, kann die Kondition unseres Problems beliebig schlecht werden, wie wir uns schnell verdeutlichen können: Würden wir die Nullfunktion als Observable wählen, finden sich – sofern ein passendes ϕ gewählt wurde – im l^∞-Sinn überabzählbar viele nahe Observablen $\hat{o}$, für die $\mathbf{\Phi}_{\phi,\hat{o}}$ eine Einbettung ist. Die korrespondierenden Trajektorien im $\mathbb{R}^{2m+1}$ verlaufen in diesem Fall jedoch sehr nahe bei 0 und das Problem der Attraktoranalyse ist schlecht konditioniert.
Man kann die Bedingungen an ϕ auch weiter konkretisieren:

> *Sei $\phi : M_0 \to M_0$ ein Diffeomorphismus, der folgende Bedingungen erfüllt:*
>
> 1. *Die Menge $X_k := \left\{\mathbf{x} \in M_0 \mid \phi^k(\mathbf{x}) = \mathbf{x}\right\}$ der ϕ^k-periodischen Punkte hat für alle $k \leq 2m$ endliche Kardinalität.*
> 2. *Für alle Punkte $\mathbf{x} \in X_k$, $k \leq 2m$ sind die Eigenwerte von $(D\phi^k)_{\mathbf{x}}$ paarweise verschieden.*
>
> *Dann bilden alle $o \in C^2(M_0, \mathbb{R})$, für die $\mathbf{\Phi}_{(\phi,o)}$ keine Einbettung ist, eine Nullmenge in $C^2(M_0, \mathbb{R})$.*

Ein einfaches Beispiel, welches nicht diesen Bedingungen genügt und auch nicht zu einer Einbettung führt, ist $\phi = \mathrm{id}$. Dies widerspricht sowohl Bedingung 1 als auch Bedingung 2.

Dass $\phi = \mathrm{id}$ nie zu einer Einbettung $\mathbf{\Phi}$ für $m \geq 2$ führen kann, ist offensichtlich, da jeder Punkt auf die Diagonale im $\mathbb{R}^{2m+1}$ abgebildet wird und somit M in $\mathbb{R}$ einbettbar sein müsste, was für m-dimensionale Mannigfaltigkeiten mit $m \geq 2$ nie der Fall sein kann.
Für eine tiefere Auseinandersetzung mit den Forderungen an ϕ sei auf [SYC91] verwiesen. Einen sehr ausführlich erklärten und detaillierten Beweis für Takens' Theorem, in welchem auch genauer auf die Wahl von ϕ und o eingegangen wird, findet man in [Huk06]. Nebenbei sei bemerkt, dass man hier auch eine alternative Formulierung von Takens' Theorem findet, in welchem die Glattheitsanforderungen an ϕ und o nicht mehr so streng sind wie in Takens' Formulierung aus [Tak81].

3.1.3 Alternative Formulierung für zeitkontinuierliche Prozesse

Bisher ist Takens' Theorem, in der Form von Theorem 3.1, lediglich auf zeitdiskrete Prozesse anwendbar. Aus der Theorie über gewöhnliche Differentialgleichungen ist allerdings bekannt, dass es sich bei ϕ_t für alle $t \in \mathbb{R}$ um einen Diffeomorphismus handelt und $\phi_{t+s} = \phi_t \circ \phi_s$ für alle $t, s \in \mathbb{R}$ gilt. Es ist also direkt ersichtlich, dass man bei einem kontinuierlichen Fluss ϕ_t durch Anwenden von Takens' Theorem mit

$$\phi := \phi_\tau, \ \tau \in \mathbb{R}^+ \tag{3.2}$$

ebenfalls eine Einbettung für solche ϕ_τ erhält, welche den Bedingungen in Unterabschnitt 3.1.2 genügen. Hierbei ist dann

$$(\phi_\tau)^k = \phi_{k \cdot \tau}.$$

Takens' gab in [Tak81] auch eine Formulierung für den kontinuierlichen Fall an:

Theorem 3.2 [Takens' Theorem (zeitkontinuierliche Delay-Einbettung)] *Sei M_0 eine kompakte m-dimensionale C^2-Mannigfaltigkeit, $\mathbf{X} : M \to TM$ ein C^2-Vektorfeld, welches einen Fluss $\phi_t : M \to M$ definiert. Sei weiterhin $o : M_0 \to \mathbb{R} \in C^2(M_0, \mathbb{R})$. Dann ist die Abbildung $\mathbf{\Phi}_{(\phi_t, o)} : M_0 \to \mathbb{R}^{2m+1}$, welche durch*

$$\mathbf{\Phi}_{(\phi_t,o)}(\mathbf{x}) := (o(\mathbf{x}), o(\phi_1(\mathbf{x})), o(\phi_2(\mathbf{x})), \ldots, o(\phi_{2m}(\mathbf{x}))) \tag{3.3}$$

*definiert ist, **im Allgemeinen** eine Einbettung.*

Dies ist die Formulierung, die wir erhalten, wenn wir in Gleichung (3.2) $\tau = 1$ setzen.
Der Ausdruck "im Allgemeinen" ist hier so zu verstehen wie im zeitdiskreten Fall, allerdings kann man die spezifischen Bestimmungen, die dort an ϕ gestellt wurden, hier als Forderungen an $\mathbf{X}$ formulieren:

> *Sei $\phi_t : M_0 \to M_0$ der Fluss zum Vektorfeld $\mathbf{X} \in C^2(M, TM)$ und es gelten folgende Bedingungen:*
>
> 1. *ϕ_t hat keine ganzzahlige Periode kleiner oder gleich $2m + 1$.*

2. *Für alle Punkte* $\mathbf{x} \in M_0$ *mit* $\mathbf{X}(\mathbf{x}) = 0$ *sind die Eigenwerte von* $(D\phi_1)_{\mathbf{x}}$ *paarweise verschieden und ungleich* 1.

Dann bilden alle $o \in C^2(M_0, \mathbb{R})$*, für die* $\mathbf{\Phi}_{(\phi_t, o)}$ *keine Einbettung ist, eine Nullmenge in* $C^2(M_0, \mathbb{R})$*.*

Noch offen wäre nun allerdings, ob die Trajektorie des durch $\mathbf{\Phi}_{(\phi_t, o)}$ definierten Prozesses dieselben Attraktoren hat, wie der ursprünglichen Prozesses auf M_0. Aufgrund der festen Wahl von Punkten der Trajektorie, welche in der Einbettung benutzt werden, ist nicht ersichtlich, ob der im $\mathbb{R}^{2m+1}$ entstehende Prozess vielleicht nur einen Teil des Attraktor-Bildes besucht. Um diese Zweifel auszuräumen, bewies Takens ein weiteres Theorem:

Theorem 3.3 [Wahl des Zeitparameters τ]
Seien M_0*,* $\mathbf{X}$ *und* ϕ_t *wie in Takens' Theorem 3.2. Sei weiter* $\mathbf{z_0} \in M_0$*. Dann sind die Attraktoren von*

1. *der durch* $\phi_t, t \in \mathbb{R}$ *und den Anfangspunkt* $\mathbf{z_0}$ *bestimmten Trajektorie und*

2. *der durch die Abbildung* $\phi_{i\cdot\tau}, \tau \in \mathbb{R}^+, i \in \mathbb{N}$ *und den Anfangspunkt* $\mathbf{z_0}$ *bestimmten Trajektorie*

im Allgemeinen *dieselben.*

"Im Allgemeinen" ist hier bezüglich des Parameters τ zu verstehen:

Die Menge aller $\tau \in \mathbb{R}^+$*, für die die Trajektorie, welche durch* $\phi_t, t \in \mathbb{R}$ *und* $\mathbf{z_0}$ *bestimmt wird, nicht denselben Attraktor hat wie die durch* $\phi_{i\cdot\tau}, i \in \mathbb{N}$ *und* $\mathbf{z_0}$ *bestimmte Trajektorie, bildet eine Nullmenge in* $\mathbb{R}^+$*.*

Gehen wir somit davon aus, einen zeitkontinuierlichen Prozess in äquidistanten Abständen durch eine Observable zu beobachten, so ist die Wahrscheinlichkeit zufällig einen Abstand zu wählen, welcher es nicht ermöglicht, den ursprünglichen Attraktor mittels Takens' Theorem 3.2 zu rekonstruieren, gleich 0.

Zusammenfassung: Abschließend kann gesagt werden, dass wir in der Praxis mit Wahrscheinlichkeit 1 einen Fall vorliegen haben, bei dem wir den Prozess bis auf Diffeomorphie genau – aufgrund einer Beobachtung durch eine Observable – rekonstruieren können.

3.1.4 Einbettung mittels Ableitungen

In [Tak81] wurde eine weitere Möglichkeit gezeigt, den Prozess durch eine Einbettung in $\mathbb{R}^{2m+1}$ zu rekonstruieren:

Theorem 3.4 [Einbettung mittels Ableitungen]
Sei M_0 *eine kompakte* m*-dimensionale* C^{2m+1}*-Mannigfaltigkeit und* $\mathbf{X} : M \to TM$ *ein*

C^{2m+1}-Vektorfeld, welches einen Fluss $\phi_t : M \to M$ definiert. Sei weiterhin $o : M_0 \to \mathbb{R} \in C^{2m+1}(M_0, \mathbb{R})$. Dann ist die Abbildung $\mathbf{\Phi}_{(\phi_t,o)} : M_0 \to \mathbb{R}^{2m+1}$, welche durch

$$\mathbf{\Phi}_{(\phi_t,o)}(\mathbf{x}) := \left(\underbrace{(o(\phi_t(\mathbf{x})))_{t=0}}_{=\, o(\mathbf{x})}, \left(\frac{\partial}{\partial t} o(\phi_t(\mathbf{x}))\right)_{t=0}, \left(\frac{\partial^2}{\partial t^2} o(\phi_t(\mathbf{x}))\right)_{t=0}, \ldots, \left(\frac{\partial^{2m}}{\partial t^{2m}} o(\phi_t(\mathbf{x}))\right)_{t=0} \right) \tag{3.4}$$

*definiert ist, **im Allgemeinen** eine Einbettung.*

"Im Allgemeinen" ist hier genauso zu verstehen wie bei Theorem 3.2. Fasst man die Ableitungen bereits diskret als finite Differenzen auf, ist die Einbettung nichts anderes als eine lineare Kombination der bekannten Delay-Einbettung.
Diese Herangehensweise hat gegenüber der Einbettung aus Theorem 3.2 den Vorteil, dass die resultierenden Punkte im $\mathbb{R}^{2m+1}$ intuitiver verstanden werden können. Jede Ableitung enspricht hier einer Raumrichtung und so lässt zum Beispiel der Betrag einer Koordinate direkt Rückschlüsse auf den eigentlichen Prozess und dessen Ableitung zu. Zudem kann man die Bewegungsgleichungen des Prozesses meistens als Differentialgleichungssystem höherer Ordnung in einer Variable auffassen. Von diesem Standpunkt aus erscheint diese Wahl der Einbettung sehr natürlich.
Ein großer Nachteil ist allerdings die höhere Glattheitsanforderung in den Voraussetzungen von Theorem 3.4. Diese sind nötig, um die Ableitungen überhaupt bilden zu können. Des Weiteren liegen in der Praxis oftmals nur diskrete Zeitreihendaten vor, und es ist unmöglich, die Ableitung auszurechnen. Diese muss dann numerisch approximiert werden. In [KS04] wird zusätzlich gezeigt, dass die Anfälligkeit für Rauschen in der Praxis sehr viel höher ist, als in der Methode aus Theorem 3.2.
Diese Einbettungsmethode sei nur der Vollständigkeit halber erwähnt. Wir werden sie im weiteren Verlauf nicht verwenden.

3.1.5 Die Einbettungsdimension $2m + 1$

Ein wenig seltsam mag die Einbettungsdimension $2m + 1$ erscheinen. Diese ist ein Resultat aus *Whitneys Einbettungssatz*, welcher besagt, dass für eine glatte, kompakte m-dimensionale Mannigfaltigkeit M im $\mathbb{R}^k$ fast jede glatte Abbildung von $\mathbb{R}^k$ nach $\mathbb{R}^{2m+1}$ eine Einbettung von M ist [SYC91]. Dieser Satz lässt sich – ähnlich zu Takens' Theorem – auch für kompakte Mengen formulieren und somit verallgemeinern.
Von diesem Einbettungssatz wird im Beweis von Takens' Theorem Gebrauch gemacht. Es ist jedoch in vielen Fällen so, dass eine Einbettung in eine Dimension $n < 2m + 1$ existiert.
Dass es für eine endliche Menge von Datenpunkten im $\mathbb{R}^{2m+1}$ auch ausreichend sein kann, diese in einen Raum kleinerer Dimensionen einzubetten ohne dabei die Attraktorstruktur zu verlieren, zeigt das Lemma von Johnson-Lindenstrauss [JL84].
Eine allgemeinere Variante von Whitneys Einbettungssatz findet sich im *Theorem von*

Menger-Nöbeling, welches aussagt, dass sich ein beliebiger m-dimensionaler, kompakter, metrischer Raum homöomorph in $\mathbb{R}^{2m+1}$ einbetten lässt.
Interessant ist, dass man die Zahl $2m + 1$ auch in *Kolmogorovs Superpositionstheorem* findet. Ein Zusammenhang zwischen diesem und dem Theorem von Menger-Nöbeling wird in [Bra09] beschrieben.

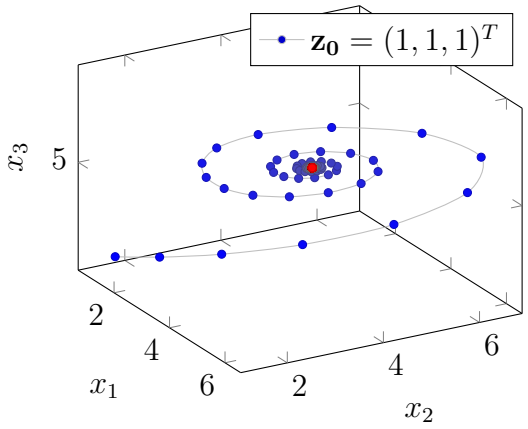
3.2 Beispiele

Beispiel 3.1: [Rekonstruktion der Trajektorien des Lorenz-Prozesses] Anhand des Beispiels 2.4 wollen wir uns das Prinzip der Delay-Einbettung veranschaulichen. Betrachtet werden nun die Rekonstruktionen der Trajektorien aus Abbildung 2.1. Die Observable o wird als

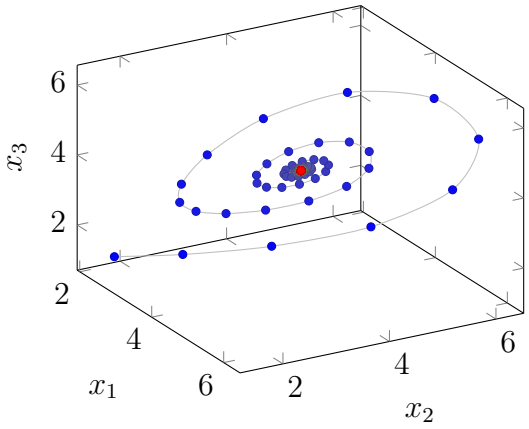
$$o((x_1, x_2, x_3)^T) = x_1$$

gewählt. $\mathbf{X}$ und o erfüllen somit die Glattheitsvoraussetzungen aus Takens' Theorem für zeitkontinuierliche Prozesse. Die Delay-Einbettung wird in $\mathbb{R}^3$ vorgenommen, da dort bereits der ursprüngliche Prozess verlief. Das Resultat in Abbildung 3.1 zeigt, dass die grundlegende Struktur der Trajektorie und des Attraktors durch die Rekonstruktion erhalten bleibt.

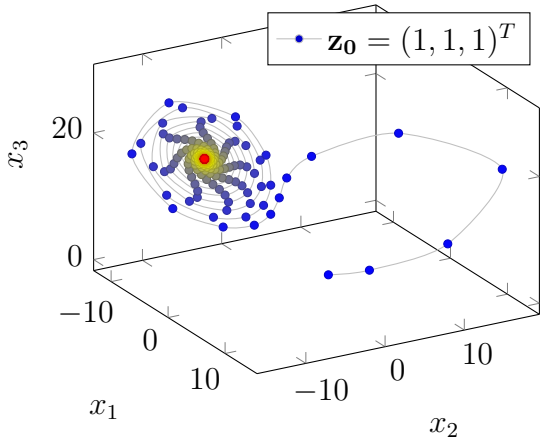
Beispiel 3.2: [Rekonstruktion der Trajektorien der Henon-Abbildung] Wir orientieren uns am Beispiel 2.5. Betrachtet werden nun die Rekonstruktionen der Trajektorien der Henon-Abbildung. Wir lassen hierbei die Parameter a, b und den Startpunkt fest und variieren lediglich die Observable o. Die Delay-Einbettung wird in $\mathbb{R}^2$ vorgenommen, da dort bereits der ursprüngliche Prozess verlief.
In den Abbildungen 3.2 (b) und (c) ist noch eine Ähnlichkeit zur Original-Struktur sichtbar. (d) scheint jedoch keine Gemeinsamkeiten mit dem ursprünglichen Attraktor-Bild zu besitzen. Es kann somit konstatiert werden, dass die Güte der Rekonstruktion maßgeblich von der Observablen abhängt.



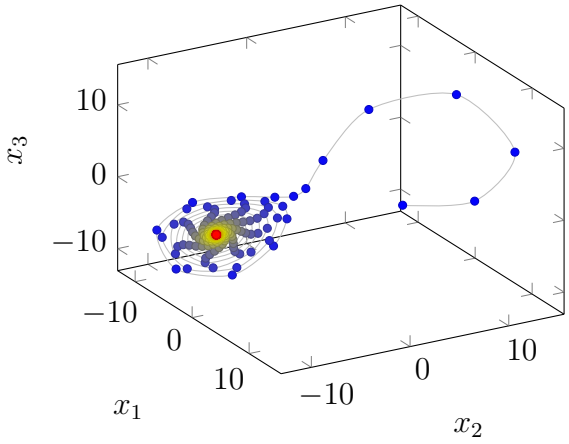
(a) Original ($b = 7$)



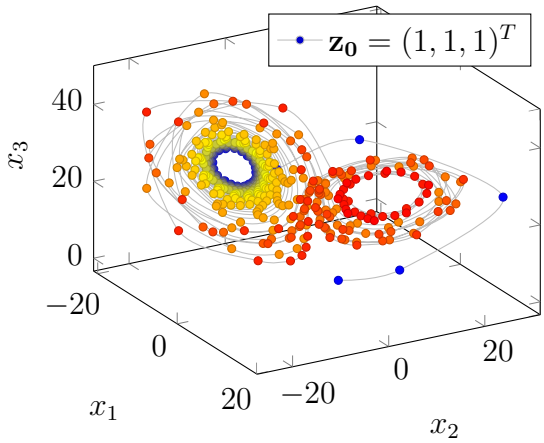
(b) Rekonstruktion ($b = 7$)



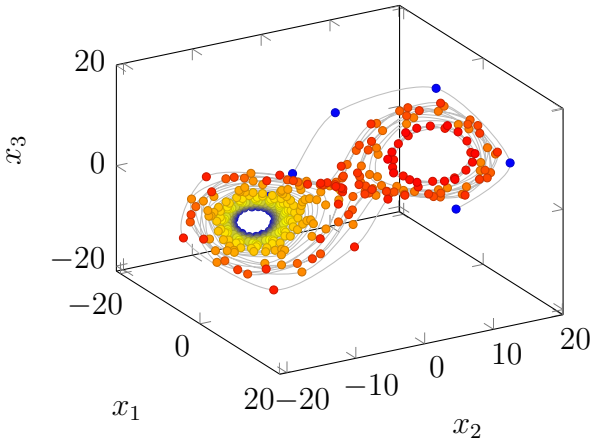
(c) Original ($b = 18$)



(d) Rekonstruktion ($b = 18$)

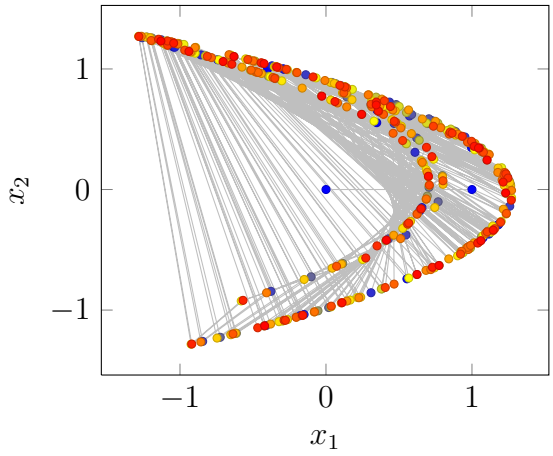


(e) Original ($b = 28$)

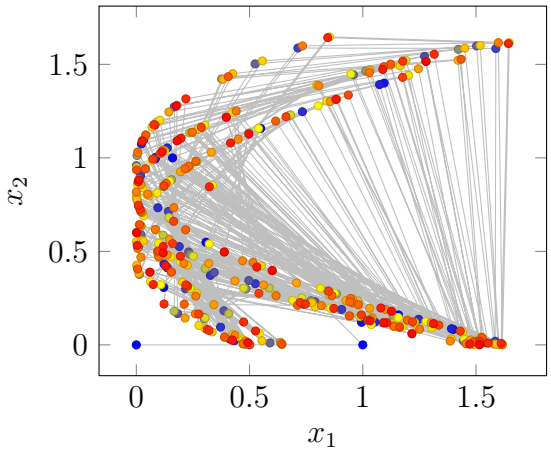


(f) Rekonstruktion ($b = 28$)

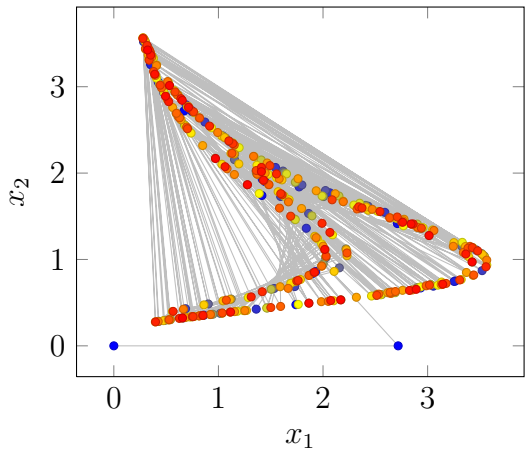
Abb. 3.1: Rekonstruktionen des Lorenz-Attraktors ($a = 10$, $c = \frac{8}{3}$)



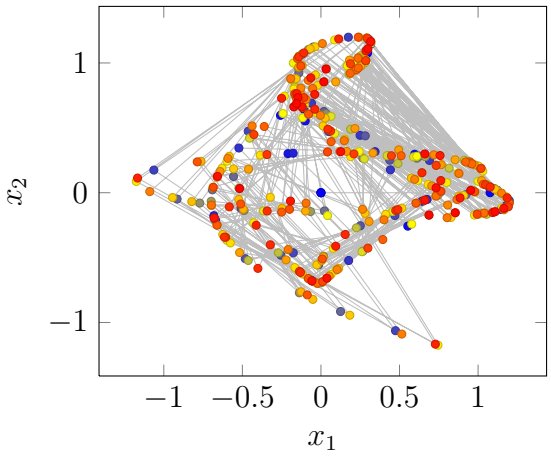
(a) Original



(b) Rekonstruktion ($o((x_1, x_2)^T) = x_2^2$)



(c) Rekonstruktion ($o((x_1, x_2)^T) = \exp(x_2)$)



(d) Rekonstruktion ($o((x_1, x_2)^T) = x_1 \cdot x_2$)

Abb. 3.2: Rekonstruktionen des Henon-Attraktors für $a = 1.4$, $b = 0.3$ und $\mathbf{z_0} = (0, 0)^T$

4 Anwendung von Takens' Theorem in der Praxis

Bisher haben wir gesehen, dass wir bei Zeitreihen, die einem zugrundeliegenden Prozess folgen (siehe Definition 2.1 und 2.2), die gesamte Trajektorie diffeomorph rekonstruieren können. Läuft diese auf einen Attraktor wie in Definition 2.5 zu, so können wir auch den Attraktor, in welchem sich der Prozess im Grenzwert $t \to \infty$ befindet, rekonstruieren – zumindest den Teil des Attraktors, der von der Trajektorie durchlaufen wird. Dies soll uns die Möglichkeit geben aufgrund der Zeitreihe den zugrundeliegenden Prozess auf dem Attraktor modellieren und vorhersagen zu können. Weiterhin haben wir gesehen, dass ein Großteil aller relevanten Prozesse in der Praxis dissipativ (siehe Definition 2.3) ist und dass diese Prozesse immer von einem Attraktor angezogen werden.
Will man das Delay-Einbettungsschema von Takens' nun in der Praxis auf eine Zeitreihe anwenden, stößt man schnell auf mehrere Probleme. Auf diese wollen wir in diesem Kapitel eingehen. In Kapitel 5 werden wir anschließend die Methoden zur Vorhersage des Prozesses und der Zeitreihe erläutern.
In diesem Kapitel orientieren wir uns vor allem an [KS04, LV07, Eng91].

4.1 Gestalt, Rauschen und Endlichkeit der Zeitreihe

Eine offensichtliche Frage ist, ob die vorliegende Zeitreihe von einem zugrundeliegenden Prozess herrührt und somit überhaupt eine Gestalt hat, wie wir sie fordern. Diese Frage ist berechtigt, jedoch ist die Herangehensweise an das Problem meist eine andere: Man analysiert die Zeitreihe nun gerade aus dem Grund, dass man hofft, einen zugrundeliegenden Prozess und die damit verbundenen Gleichungssysteme zu finden und modellieren zu können. Somit geht man a-priori davon aus, dass ein solches System existiert. Sind die Vorhersagen und das Modell, welches man erhält, jedoch nicht mit Beobachtungen oder anderen Experimenten vereinbar, so kann man sich die Frage stellen, ob das System möglicherweise eine stochastische Gestalt hat und somit nicht mehr deterministisch ist oder sich das zugrundeliegende Gleichungssystem vielleicht mit der Zeit ändert. Letzteres führt oftmals dazu, dass sich die Gestalt eines Attraktors verändern kann und Trajektorien plötzlich einen vollkommen unerwarteten Verlauf aufweisen. Diesen Effekt nennt man in der Literatur *Bifurkation*. Für eine ausführliche Behandlung des Themas sei auf [KS04, Kus04] verwiesen.
Als heuristischen Test, ob man davon ausgehen kann, dass eine Zeitreihe deterministische

Struktur hat, bieten sich sogenannte *Surrogate-Tests* an. Hier ermittelt man zunächst eindeutige Kenngrößen der Zeitreihe (wie z.B. das diskrete Fourier-Spektrum) und konstruiert dann einfache Prozesse (z.B. *stochastische Autoregressive-Moving-Average*-Prozesse (*ARMA*)), die ebenfalls die gleichen Kenngrößen aufweisen. Kann man mit diesen Prozessen den Verlauf der eigentlichen Zeitreihe gut simulieren, geht man davon aus, dass die stochastische Komponente in etwa dieselbe wie im ARMA-Prozess ist. Auch hierzu findet sich eine detaillierte Beschreibung in [KS04].
Neben der stochastischen Komponente im Prozess ist in der Praxis zusätzlich die Zeitreihe verrauscht. Dies geschieht z.B. durch Messungenauigkeiten oder die begrenzte Darstellungsmöglichkeit am Rechner. Geeignete Methoden zur Reduzierung des Rauschens auf bereits vorliegenden Daten (sogenannte *Noise-Filter)* lassen sich ebenfalls in [KS04] finden.
Ein Problem, welches man in der Theorie nicht beachtet, ist die Tatsache, dass Zeitreihen in der Praxis immer endlich sind. Es liegt eine endliche Anzahl von Messungen und Beobachtungen vor. Selbst wenn man eine unendliche Zeitreihe vorliegen hätte, bliebe das Problem, diese im Rechner zu verarbeiten. Hieraus resultiert ein klares Problem bei der Anwendung von Takens' Theorem: Wollen wir nun den Attraktor rekonstruieren, so ist dies in der Theorie möglich, da sich die Trajektorie nach genügend langer Zeit beliebig nahe am Attraktor bewegt. Es ist jedoch unklar, wie lange wir warten müssen, bis die Trajektorie diesen annähernd erreicht. Wie wir in Satz 2.4 bewiesen haben, hat der Attraktor eines dissipativen Prozesses Lebesgue-Maß 0. Die Wahrscheinlichkeit, dass wir somit mit einer zufällig gewählten Anfangsbedingung bereits auf dem Attraktor liegen, ist ebenfalls gleich 0. Es wird eine Phase der *Transienz* geben, bevor der Prozess annähernd den Bewegungsgleichungen auf dem Attraktor folgt. Jedoch sind es genau diese, die wir rekonstruieren wollen.

Bemerkung: Die Ungleichungen aus Satz 2.4 erlauben uns im Allgemeinen keine Schätzung für eine adäquate Wartezeit, bis das System den Attraktor erreicht hat. Es liegt zwar die exponentielle Verkleinerung des Phasenraumvolumen vor, jedoch ist unklar, ob der Attraktor bereits erreicht wurde, sobald das Volumen auf 0 geschrumpft ist. Zudem ist es in der Praxis nicht ohne Weiteres möglich die Konstanten c_1 und c_2 zu schätzen.

4.2 Wahl der Einbettungsdimension

Ein weiteres Problem, mit welchem wir uns bisher nicht beschäftigt haben, ist die Mannigfaltigkeitendimension m. Wie wir bereits im Unterabschnitt 3.1.1 erfahren haben, reicht es, hier die *Boxcounting-Dimension* des Attraktors zugrunde zu legen. Da diese aber nicht a-priori bekannt ist, benötigen wir Methoden, um sie schätzen. Zunächst werden verschiedene Dimensionsbegriffe erläutert und deren Eignung als Dimensionsschätzer für Attraktoren anhand einer gegebenen Punktmenge diskutiert. Im Anschluss werden die Probleme der Einbettung von Zeitreihen in hohe Dimensionen skizziert.

4.2.1 Hausdorff-Dimension

Bisher kennen wir den Dimensionsbegriff z.B. aus Vektorräumen, in denen die minimale Anzahl an aufspannenden Vektoren als Dimension bezeichnet wurde. Bei Mannigfaltigkeiten bezieht sich die Dimension auf den reellen Vektorraum, durch welchen lokal mittels Karten die Mannigfaltigkeit dargestellt wird. Diese Begriffe haben die Gemeinsamkeit, dass die betreffenden Dimensionen ganzzahlig sind. Oftmals sind Attraktoren jedoch so komplex, dass ein solcher Dimensionsbegriff nicht genügt, um sie ausreichend zu beschreiben. Hier benötigen wir *fraktale* Dimensionen, also solche, die reelle Werte annehmen. Der wohl bekannteste Dimensionsbegriff dieser Art ist die nach Felix Hausdorff benannte *Hausdorff-Dimension* aus der Maßtheorie:

Definition 4.1 [Hausdorff-Dimension]
Sei $X \subset M$ eine Teilmenge einer m-dimensionalen Riemannschen Mannigfaltigkeit. Wir definieren das s-dimensionale Hausdorffmaß für beliebiges $s \in \mathbb{R}^+$ durch

$$H^s(X) := \lim_{\epsilon \searrow 0} \inf_{(X_i)_{i=1}^{\infty}} \left\{ \sum_{i=1}^{\infty} d(X_i)^s \mid X \subseteq \bigcup_{i=1}^{\infty} X_i;\ d(X_i) < \epsilon \right\}. \tag{4.1}$$

*d bezeichnet hier den durch die Riemannsche Metrik induzierten Durchmesser einer Menge. Das Infimum ist über alle abzählbaren Überdeckungen von X durch Mengen X_i zu bilden. Die **Hausdorff-Dimension** von X ist nun*

$$\dim_H(X) := \inf \left\{ s \in \mathbb{R}^+ \mid H^s(X) = 0 \right\} = \sup \left\{ s \in \mathbb{R}^+ \mid H^s(X) = \infty \right\}. \tag{4.2}$$

Aus der Maßtheorie ist bekannt, dass man sich bei den überdeckenden Mengen X_i auf m-dimensionale Würfel beschränken kann und trotzdem dasselbe Maß erhält. Für Vektorräume und Mannigfaltigkeiten ist die Hausdorff-Dimension die gleiche wie die durch unseren ursprünglichen Dimensionsbegriff gegebene. Des Weiteren ist aus der Maßtheorie bekannt, dass das Lebesguemaß im $\mathbb{R}^d$ ein Vielfaches von H^d ist. Somit gilt also für Lebesgue-Nullmengen A, dass sie auch Hausdorff-Nullmengen bzgl. H^d sind. Also gilt für Attraktoren von dissipativen Prozessen:

$$\dim_H(A) \leq \dim_H(M)$$

Dies folgt ohnehin sofort aus der Definition, da $A \subset M$ ist und eine Überdeckung von M auch A überdeckt. Dass oftmals auch

$$\dim_H(A) < \dim_H(M)$$

gilt, lässt sich erahnen, da $H^{\dim_H(M)}(M) = \dim_H(M)$ ist und wegen des Zusammenhangs zwischen Lebesgue- und Hausdorffmaß $H^{\dim_H(M)}(A) = 0$ gilt. Den Fall der echt kleineren Dimension beobachtet man bei allen praktisch relevanten Beispielen.
Ein großer Nachteil für die praktische Berechnung der Hausdorff-Dimension ist die Tat-

sache, dass die Überdeckungsmengen auch jeweils verschiedene Durchmesser besitzen können. Eine numerische Approximation ist hier aussichtslos. Will man die Hausdorff-Dimension schätzen, verwendet man in der Praxis meistens Schätzer der Boxcounting-Dimension, die wir in Unterabschnitt 4.2.2 einführen werden.

4.2.2 Renyi-Dimension der Ordnung q

Der Begriff der q-Dimension, bzw. *Renyi-Dimension der Ordnung* q geht zurück auf Alfred Renyi [Ren59]. Die Idee ist hierbei, die betreffende Menge $X \subset M$ mit gleichgroßen Würfeln zu überdecken und die Anzahl der Würfel zu zählen, deren Schnitt mit X nicht leer ist. Im Grenzwert lässt man dann die Kantenlänge der Würfel gegen 0 gehen.
Die hier gegebenen Definitionen orientieren sich an [LV07].

Definition 4.2 [Renyi-Dimension der Ordnung q]
Sei $X \subset M$ eine beschränkte μ-meßbare Teilmenge einer m-dimensionalen Riemannschen Mannigfaltigkeit. μ ist hierbei ein beliebiges Wahrscheinlichkeitsmaß auf M. Sei weiter $q \in \mathbb{R}^+$ beliebig. Man überdecke die Mannigfaltigkeit mit einem regulären Gitter bestehend aus Würfeln der Kantenlänge ϵ. Daraus resultieren $Z(\epsilon)$ viele Würfel, die einen Schnitt mit X haben, der ein Maß echt größer 0 hat. Diese indiziere man beliebig: $(W_i)_{i=1}^{Z(\epsilon)}$. Es seien

$$\begin{aligned} D_q^-(X) &:= \liminf_{\epsilon \searrow 0} \frac{\log \sum_{i=1}^{Z(\epsilon)} \mu(W_i \cap X)^q}{(q-1)\log \epsilon}, \\ D_q^+(X) &:= \limsup_{\epsilon \searrow 0} \frac{\log \sum_{i=1}^{Z(\epsilon)} \mu(W_i \cap X)^q}{(q-1)\log \epsilon} \ . \end{aligned}$$

Falls $D_q^-(X) = D_q^+(X)$ ist, nennen wir

$$D_q(X) := D_q^+(X) = D_q^-(X) = \lim_{\epsilon \searrow 0} \frac{\log \sum_{i=1}^{Z(\epsilon)} \mu(W_i \cap X)^q}{(q-1)\log \epsilon} \tag{4.3}$$

*die **Renyi-Dimension der Ordnung** q von X.*

Das Maß μ soll hierbei die Verteilung der Trajektorien auf dem Attraktor A widerspiegeln. Es gibt an, wie wahrscheinlich es ist, in einem Bereich von A gerade eine Trajektorie vorzufinden.
Eine Voraussetzung hierfür ist z.B.

$$\mu(M \setminus A) = 0 \ , \tag{4.4}$$

da wir Trajektorien auf dem Attraktor messen wollen, wo sie sich unendlich lange aufhalten. Dass wir dennoch $\mu(W_i \cap X)$ statt $\mu(W_i)$ schreiben, hat den Grund, dass wir den Begriff möglichst allgemein halten wollen und nicht explizit (4.4) fordern. Die Existenz

der Renyi-Dimension setzt voraus, dass der Grenzwert existiert – also Limes Superior und Limes Inferior denselben Wert annehmen. Dies ist z.B. für die Einschränkung auf das Lebesgue-Maß und Borel-meßbare Mengen immer der Fall. Für allgemeine Wahrscheinlichkeitsmaße muss dies jedoch nicht gelten, siehe [Ren59]. Wir gehen im Folgenden immer davon aus, dass die Renyi-Dimension existiert.
Durch das Maß μ bekommen wir bei diesem Dimensionsbegriff die Möglichkeit, Gebiete des Attraktors, in welchem sich der Prozess öfter aufhält als in anderen, stärker zu gewichten.
Spezialfälle der Renyi-Dimension sind z.B. die Boxcounting-Dimension und die Korrelationsdimension, die wir nun näher betrachten wollen.

Boxcounting-Dimension (Renyi-Dimension der Ordnung 0)

Definition 4.3 [Boxcounting-Dimension]
Die Renyi-Dimension der Ordnung 0 *nennt man* ***Boxcounting-Dimension*** *oder auch* ***Kapazitätsdimension****.*

Es gilt also

$$\begin{aligned} d_{cap}(X) := D_0(X) &= \lim_{\epsilon \searrow 0} \frac{\log \sum_{i=1}^{Z(\epsilon)} \mu(W_i \cap X)^0}{-\log \epsilon} && (4.5) \\ &= \lim_{\epsilon \searrow 0} \frac{\log \sum_{i=1}^{Z(\epsilon)} 1}{-\log \epsilon} && (4.6) \\ &= \lim_{\epsilon \searrow 0} \frac{\log Z(\epsilon)}{-\log \epsilon} \ . && (4.7) \end{aligned}$$

Nun sehen wir, wieso man die Renyi-Dimension der Ordnung 0 auch "Boxcounting-Dimension" nennt: Der Term im Zähler von (4.7) zählt alle Würfel, deren Schnitt mit X keine μ-Nullmenge ist.
Man sieht sofort, dass sich diese Dimension für eine endliche Anzahl N unabhängig nach μ gesampleter Punkte – und diesen Fall haben wir im Endeffekt vorliegen – sehr gut heuristisch schätzen lässt, indem man jeden Würfel zählt, welcher von mindestens einem Punkt besetzt ist. Eine Aussage über die Konvergenz zu treffen ist dennoch schwierig. In [Hun90] wird durch eine genauere Analyse gezeigt, dass man für ein festes ϵ eine Konvergenz des Schätzers gegen $Z(\epsilon)$ mit einer Rate von $O(N)$ erwarten kann. Die Konstante in dieser Abschätzung hängt allerdings insbesondere von den Wahrscheinlichkeiten $\mu(W_i \cap X)$ ab. Existieren Boxen, für welche diese Wahrscheinlichkeit klein ist, wird dies die Konvergenz verlangsamen. Hierzu sei auf Kapitel 2 in [Hun90] verwiesen.
Des Weiteren ist anzumerken, dass die $\mathbf{x_i}$ durch zeitliche Korrelationen der Samples nicht mehr unabhängig sind. Wie man diese Abhängigkeiten minimiert, wird in Abschnitt 4.3 erklärt.
Es sei nochmals darauf hingewiesen, dass die oben beschriebene Konvergenz

1. lediglich für ein festes ϵ festzustellen ist und

2. nur mit einer von μ und ϵ abhängigen Konstante gilt.

Somit sollte man dieses Resultat keineswegs überbewerten.

Bemerkung: Will man anhand endlich vieler Samples die Hausdorff-Dimension schätzen, so benutzt man einen Boxcounting-Schätzer. Dies ist nicht weiter verwunderlich, wenn man sich verdeutlicht, wie diese beiden Begriffe zusammenhängen. Besonders empfohlen sei hierzu [Mai93].

Wir wollen kurz skizzieren, dass es hier einen engen Zusammenhang gibt: Würde man in der Definition des Hausdorff-Maßes aus Gleichung (4.1) – statt beliebigen Überdeckungsmengen mit einem durch ϵ beschränkten Durchmesser – ausschließlich Würfel zulassen, welche alle denselben Durchmesser ϵ haben, so erhielte man:

$$H_*^s(X) := \lim_{\epsilon \searrow 0} \left(Z(\epsilon) \cdot \epsilon^s \right).$$

Hierzu müsste man formal noch beweisen, dass – für ϵ gegen 0 – keine "günstigere" Überdeckung durch Würfel existiert als die Anordnung dieser in einem regelmäßigen Gitter. Darauf wird hier verzichtet.
Bei der Boxcounting-Dimension wird gefordert, dass sich die Anzahl okkupierter Würfel für ϵ gegen 0 asymptotisch wie eine negative Potenz von ϵ verhält (siehe (4.7)):

$$Z(\epsilon) \overset{\epsilon \to 0}{\sim} \epsilon^{-d_{cap}}$$

Damit das modizfizierte Hausdorff-Maß H_*^s bei d_{cap} einen Sprung zwischen 0 und ∞ macht, müssen sich die beiden Terme genauso verhalten. Es muss gelten:

$$\begin{aligned} Z(\epsilon) &\overset{\epsilon \to 0}{\to} \infty \quad \text{schneller als} \quad \epsilon^s \overset{\epsilon \to 0}{\to} 0 \quad \text{für} \quad s < d_{cap} \quad \text{und} \\ Z(\epsilon) &\overset{\epsilon \to 0}{\to} \infty \quad \text{langsamer als} \quad \epsilon^s \overset{\epsilon \to 0}{\to} 0 \quad \text{für} \quad s > d_{cap}. \end{aligned}$$

Dies skizziert grob, warum die Boxcounting-Dimension eine Art Vereinfachung der Hausdorff-Dimension ist: Als Überdeckungselemente werden nur gleichgroße Würfel zugelassen.

Korrelationsdimension (Renyi-Dimension der Ordnung 2)

Definition 4.4 [Korrelationsdimension]
Die Renyi-Dimension der Ordnung 2 nennt man ***Korrelationsdimension*** *oder auch* ***Grassberger-Procaccia-Dimension****.*

Eine einfache Formel für die anhand eines endlichen Samples geschätzte Korrelationsdimension kann man finden, wenn man in der Definition der Renyi-Dimension 4.2 die Mannigfaltigkeit M nicht mit Würfeln, sondern in ähnlicher Weise mit Kugeln überdeckt.

Dies führt im Grenzwert zu einem äquivalenten Dimensionsbegriff (siehe [HP83]):

$$D_q(X) := D_q^+(X) = D_q^-(X) = \lim_{\epsilon \searrow 0} \frac{\log \int_X \mu\left(\overline{B_\epsilon}(\mathbf{x}) \cap X\right)^{q-1} d\mu(\mathbf{x})}{(q-1)\log \epsilon}. \tag{4.8}$$

$\overline{B_\epsilon}(\mathbf{x})$ ist hier der abgeschlossene ϵ-Ball um $\mathbf{x}$:

$$\overline{B_\epsilon}(\mathbf{x}) := \{\mathbf{y} \in M \mid ||\mathbf{x} - \mathbf{y}||_2 \leq \epsilon\}.$$

Hat man nun N Samples $\mathbf{x_i} \in X$ vorliegen, kann man das Integral im Zähler durch eine Summe abschätzen. Lassen wir dann N beliebig groß werden, erhalten wir folgenden Ausdruck für $q = 2$:

$$d_{cor} := \lim_{\epsilon \searrow 0} \lim_{N \to \infty} \frac{\log\left(\frac{2}{N(N-1)} \cdot \sum_{i=1}^{N} \sum_{j=i+1}^{N} H\left(\epsilon - ||\mathbf{x_i} - \mathbf{x_j}||_2\right)\right)}{\log \epsilon}. \tag{4.9}$$

H stellt die *Heavyside*-Funktion

$$H(z) := \begin{cases} 0 & \text{falls } z < 0 \text{ und} \\ 1 & \text{sonst} \end{cases}$$

dar. Diese repräsentiert hierbei die Kugel in der Überdeckung. Für jeden Punkt $\mathbf{x_i}$ werden alle Punkte gezählt, die in der ϵ-Kugel um $\mathbf{x_i}$ liegen.

Satz 4.5
Für ein nach μ verteiltes, unabhängiges Sampling konvergiert der Zähler aus (4.9) *mit einer Rate von $O(N^{\frac{1}{2}-\epsilon})$ schwach gegen den Zähler in* (4.8) *für $q = 2$. $\epsilon > 0$ kann hierbei beliebig klein gewählt werden.* [1]

Beweis. Es gilt

$$\int_X \int_X H\left(\epsilon - ||\mathbf{x} - \mathbf{y}||_2\right) d\mu(\mathbf{x}) d\mu(\mathbf{y}) = \int_X \mu(B_\epsilon(\mathbf{x}) \cap X) d\mu(\mathbf{x}). \tag{4.10}$$

Somit handelt es sich bei

$$\frac{1}{N(N-1)} \sum_{i=1}^{N} \sum_{\substack{j=1 \\ j \neq i}}^{N} H\left(\epsilon - ||\mathbf{x_i} - \mathbf{x_j}||_2\right)$$

[1] Dieser Beweis lässt sich bei Verwendung von (4.8) mit $q = 0$ auch für einen alternativen Boxcounting-Schätzer formulieren. Allerdings muss für die Anwendung eines solchen Schätzer der gesamte Attraktor bereits so genau abgetastet sein, dass für alle $\mathbf{x_i}$ mindestens ein $\mathbf{x_j}$, $j \neq i$ existiert, sodass $H(\epsilon - ||\mathbf{x_i} - \mathbf{x_j}||_2)^{-1} = 1$ ist.

um eine *U-Statistik*, welche nach [Hoe48] ein erwartungstreuer Schätzer für (4.10) ist. Da

$$\forall\, r \in \mathbb{N}: \int_X \int_X H(\epsilon - ||\mathbf{x}-\mathbf{y}||_2)^r d\mu(\mathbf{x}) d\mu(\mathbf{y}) \leq 1 < \infty$$

gilt, lässt sich auf diese U-Statistik das Theorem 3.1 aus [GS73] anwenden und die Behauptung folgt. □

Somit ist gezeigt, dass die Rate $\sqrt{N}$ beliebig angenähert werden kann.
Es sei angemerkt, dass in Theorem 4.1 in [GS73] auch eine fast sichere Konvergenz mit einer Rate von $O(N^{2r})$ für beliebige natürliche Zahlen r hergeleitet wird. Man übertrifft somit jede polynomielle Rate. Bei genauerer Betrachtung allerdings zeigt sich, dass die Konstante in dieser Abschätzung die Größenordnung δ^{2r} hat, wobei δ mit dem Fehler der Abschätzung zusammenhängt.
In [MW94] wird erwähnt, dass für den Term

$$\frac{1}{N^2} \sum_{i,j=1}^{N} H(\epsilon - ||\mathbf{x_i} - \mathbf{x_j}||_2) \tag{4.11}$$

auch eine fast sichere Konvergenz mittels des Ergodensatzes gezeigt werden kann. Dies ist auf die uniforme Konvergenz der empirischen Verteilung

$$\frac{1}{N} \sum_{i=1}^{N} \delta_{\mathbf{x_i}} \overset{N\to\infty}{\longrightarrow} \mu$$

zurückzuführen. Der Schätzer aus (4.11) ist allerdings nur asymptotisch erwartungstreu, da für nach μ gezogene unabhängige Zufallsvariablen $\mathbf{X_i}, i = 1, \ldots, N$

$$\begin{aligned}
& \mathbb{E}\left[\frac{1}{N^2} \sum_{i,j=1}^{N} H(\epsilon - ||\mathbf{X_i} - \mathbf{X_j}||_2)\right] \\
= \; & \mathbb{E}\left[\frac{N-1}{N} \cdot \frac{1}{N(N-1)} \left(\sum_{i=1}^{N} \sum_{\substack{j=1 \\ j\neq i}}^{N} H(\epsilon - ||\mathbf{X_i} - \mathbf{X_j}||_2) + N\right)\right] \\
= \; & \frac{1}{N} + \frac{N-1}{N} \cdot \mathbb{E}\left[\frac{1}{N(N-1)} \sum_{i=1}^{N} \sum_{\substack{j=1 \\ j\neq i}}^{N} H(\epsilon - ||\mathbf{X_i} - \mathbf{X_j}||_2)\right] \\
= \; & \frac{1}{N} + \frac{N-1}{N} \cdot \int_X \mu(B_\epsilon(\mathbf{x}) \cap X) d\mu(\mathbf{x})
\end{aligned}$$

gilt. Dieser Schätzer konvergiert aber mit einer Rate von $O(N)$ gegen unseren Schätzer

in (4.9) und es kann somit von einer Konvergenzrate von $O(\sqrt{N})$ für die fast sichere Konvergenz unseres Schätzers ausgegangen werden. Die Rate $O(\sqrt{N})$ ist hierbei auf die Konvergenzgeschwindigkeit im Ergodensatz, bzw. im Gesetz der großen Zahlen zurückzuführen.
Wie man sieht, stammt der Name "Korrelationsdimension" daher, dass man räumliche Korrelationen zwischen verschiedenen Punkten der Menge X untersucht. Da diese im Rahmen der Zeitreihenanalyse Punkte der Trajektorie auf dem Attraktor sind, misst man also eine räumliche Korrelation zwischen zeitlich versetzten Punkten der Trajektorie.
Der Name "Grassberger-Procaccia-Dimension" ist darauf zurückzuführen, dass der diskrete Schätzer über die Korrelationssumme aus Gleichung (4.9) von Peter Grassberger und Itamar Procaccia vorgestellt wurde [GP83].
Wie bereits zu erahnen ist, eignet sich dieser Dimensionsbegriff besonders gut für eine numerische Berechnung anhand von gegebenen Samples, da Gleichung (4.9) nur vom Abstand der Samples zueinander abhängt. Näheres hierzu folgt in Unterabschnitt 4.2.6.

4.2.3 Zusammenhang der Dimensionsbegriffe

Die hier eingeführten Begriffe der Boxcounting-Dimension ($q = 0$) und Korrelationsdimension ($q = 2$) sind Spezialfälle der Renyi-Dimension. Diese kann man für beliebige $q \in \mathbb{R}^+$ definieren, allerdings ist das Entwickeln eines geeigneten Dimensionsschätzer meistens sehr schwierig, weshalb wir uns auf die hier genannten Spezialfälle beziehen.
Eine Eigenschaft der Renyi-Dimensionen ist die Tatsache, dass folgende Ungleichung gilt (siehe [Ott93]):

$$D_q(X) \leq D_{q'}(X) \quad \text{für } q' \leq q. \tag{4.12}$$

Somit gilt also

$$d_{cor} \leq d_{cap}. \tag{4.13}$$

Dies ist für Dimensionsschätzer in der Praxis relevant. Das Ergebnis in [SYC91] besagt, dass wir bei einem Attraktor der Boxcounting-Dimension d_{cap} eine Einbettung nach Takens in $\mathbb{R}^{\lceil 2d_{cap}+1 \rceil}$ vornehmen können. Verwenden wir zum Schätzen der Dimension von A einen Schätzer der Korrelationsdimension, so kann es passieren, dass wir zwar die Korrelationsdimension gut approximieren, diese aber soviel kleiner als die Kapazitätsdimension des Attraktors ist, dass wir den Fall

$$\lceil 2d_{cap} + 1 \rceil > \lceil 2d_{cor} + 1 \rceil$$

vorliegen haben, was dazu führen kann, dass wir keine echte Einbettung mehr erhalten.
Des Weiteren ist anzumerken, dass immer

$$\dim_H \leq d_{cap}$$

gilt. Da wir bereits in Unterabschnitt 4.2.2 gesehen haben, dass die Boxcounting-Dimension eine Art eingeschränkte Hausdorff-Dimension ist, würden wir auch erwarten, dass in vie-

len einfachen Fällen Gleichheit der beiden Dimensionen gilt. Dies ist in der Praxis auch für komplizierte Gebilde sehr oft korrekt. Lediglich einige konstruierte Fälle weisen hier Ungleichheit auf, siehe [Mai93].

4.2.4 Eine andere Herangehensweise: Die Hauptachsenzerlegung

Bisher haben wir neue Dimensionsbegriffe und Ideen für Dimensionsschätzer kennengelernt, welche eingesetzt werden können, um anhand einer endlichen Anzahl von Punkten auf dem Attraktor dessen Dimension m zu schätzen und ihn dann in $\mathbb{R}^{\lceil 2m+1 \rceil}$ einzubetten. Hier haben wir dann die Absicht, die Trajektorien des rekonstruierten Prozesses genauer zu analysieren, um etwas über den ursprünglichen Prozess zu erfahren.
Eine etwas andere Herangehensweise liefert die *Hauptachsenzerlegung*, bzw. *Principal Component Analysis (PCA)*. Die folgenden Betrachtungen sind an [BK85] angelehnt. Für eine ausführlichere Betrachtung der PCA sowie einen Beweis, dass die Anwendung des Verfahrens immer die maximale Varianz in den Daten erhält, wird [LV07] empfohlen.
Zunächst nehmen wir an, dass ein $n \in \mathbb{N}$ vorliegt, von welchem wir wissen, dass

$$n \geq \lceil 2m + 1 \rceil$$

gilt. Dann verwenden wir die Delay-Einbettung aus Kapitel 3 und betten die vorliegende Zeitreihe in $\mathbb{R}^n$ ein. Nun gehen wir davon aus, dass die entstehenden Vektoren $\mathbf{x_i}, i = 1, \ldots, N$ nicht den ganzen Raum ausfüllen, sondern sich ausschließlich auf einem Teil des Attraktors A bewegen. Diese Untermenge des Attraktors liegt in einem affinen Unterraum $\hat{T} \subset \mathbb{R}^n$. Ist μ die Verteilung auf dem Attraktor und $\mathbf{X}$ eine Zufallsvariable auf diesem, so gilt

$$\hat{T} = \mathbb{E}_\mu[\mathbf{X}] + \operatorname{span}\{\mathbf{x_i} - \mathbb{E}_\mu[\mathbf{X}] \mid i = 1, \ldots, N\}.$$

Wir betrachten nun den zugehörigen linearen Unterraum

$$T = \hat{T} - \mathbb{E}_\mu[\mathbf{X}].$$

Wenn wir nun annehmen, dass wir genügend Samples haben, um den gesamten Attraktor gut abzudecken, dann liefert

$$n' := \dim(T) \leq n$$

eine obere Schranke für die minimale Einbettungsdimension – also die kleinste Dimension, welche noch eine Einbettung des Prozesses liefert.
Um nun n' zu berechnen, wollen wir die maximale Anzahl linear unabhängiger Vektoren in T finden. Hierzu eignet sich die PCA.

Die Hauptachsenzerlegung auf unverrauschten Daten

Da wir die Verteilung auf dem Attraktor nicht kennen, approximieren wir den Erwartungswert durch das empirische Mittel, welches wir von den Daten abziehen, damit diese

um den Koordinatenursprung zentriert sind:

$$\hat{\mathbf{x_i}} := \mathbf{x_i} - \frac{1}{N}\sum_{j=1}^{N} \mathbf{x_j}.$$

Dann schreiben wir alle N Vektoren in eine Matrix

$$\mathbf{X} \equiv \frac{1}{\sqrt{N}} \begin{pmatrix} \hat{\mathbf{x_1}}^T \\ \hat{\mathbf{x_2}}^T \\ \vdots \\ \hat{\mathbf{x_N}}^T \end{pmatrix}.$$

Nun gilt für das Bild von $\mathbf{X}^T$, dass es gerade der gesuchte Span

$$\mathrm{im}(\mathbf{X}^T) = T,$$

ist, da die Spalten von $\mathbf{X}^T$ die Vektoren $\mathbf{x_i}$ sind. Dies bedeutet, dass wir die Dimension des Bildes – also den Rang von $\mathbf{X}^T$ – berechnen wollen. Wie wir aus der linearen Algebra wissen, gilt

$$\mathrm{rg}(\mathbf{X}^T) = \mathrm{rg}(\mathbf{X}) = \mathrm{rg}(\mathbf{X}^T\mathbf{X}) \ .$$

Wir berechnen also die symmetrische $n \times n$ Kovarianzmatrix $\mathbf{C} = \mathbf{X}^T\mathbf{X}$ und wollen ihren Rang bestimmen. Dies ist jedoch denkbar einfach, da wir eine reelle, symmetrische, positiv semidefinite Matrix vorliegen haben und diese also mit reellen, nichtnegativen Eigenwerten und orthonormalen Eigenvektoren diagonalisierbar ist:

$$\mathbf{C} = \mathbf{V}\,\mathrm{diag}(\sigma_1, \ldots, \sigma_n)\mathbf{V}^T. \tag{4.14}$$

Nun betrachten wir, wieviele der σ_i echt größer 0 sind und erhalten somit

$$n' = \#\left\{i \in \{1, \ldots, N\} \mid \sigma_i > 0\right\}. \tag{4.15}$$

Wir sind jedoch noch nicht am Ziel, da wir noch die eingebetteten Vektoren in $\mathbb{R}^{n'}$ benötigen. Es liegt nun nahe, die Zeitreihe – analog zu Takens' Theorem – mittels der Delay-Einbettung in $\mathbb{R}^{n'}$ einzubetten. Wir können jedoch nicht mit Sicherheit sagen, ob dies wirklich eine Einbettung des Prozesses liefert, da wir mit der PCA lediglich sichergestellt haben, dass es einen n'-dimensionalen Unterraum gibt, der unsere Voraussetzungen erfüllt. Dies ist allerdings der Raum, der von den Eigenvektoren aufgespannt wird, welche zu echt positiven Eigenwerten von $\mathbf{X}^T\mathbf{X}$ gehören.
Um also Vektoren im $\mathbb{R}^{n'}$ zu erhalten, die eine Einbettung des ursprünglichen Prozesses darstellen, müssen wir die Vektoren im $\mathbb{R}^n$, welche wir durch die Delay-Einbettung gewonnen haben, in die Eigenbasis der betreffenden Eigenvektoren transformieren. Aus

der Darstellung in Gleichung (4.14) erhalten wir

$$\begin{aligned} \mathbf{X}^T\mathbf{X} &= \mathbf{V}\,\mathrm{diag}(\sigma_1,\ldots,\sigma_n)\mathbf{V}^T \\ \Rightarrow (\mathbf{XV})^T(\mathbf{XV}) &= \mathrm{diag}(\sigma_1,\ldots,\sigma_n). \end{aligned}$$

Wir sehen, dass $\mathbf{XV}$ die Transformation der Vektoren $\mathbf{x_i}, i = 1,\ldots,N$ in die neue Basis bezüglich der Eigenvektoren $\mathbf{v_j}, j = 1,\ldots,n$ darstellt. Die $\mathbf{v_j}$ sind hierbei die Spalten von $\mathbf{V}$. Nehmen wir nun o.B.d.A. an, dass die Eigenwerte dem Betrag nach absteigend sortiert sind, so gilt

$$\sigma_i = 0 \Leftrightarrow i \in \{n'+1,\ldots,n\}.$$

Somit erhalten wir die gewünschten Vektoren im $\mathbb{R}^{n'}$ als

$$\tilde{\mathbf{x}}_\mathbf{i} = \left(\mathbf{x_i}^T \cdot \mathbf{v_1},\ldots,\mathbf{x_i}^T \cdot \mathbf{v_{n'}}\right) \ . \tag{4.16}$$

Dies liefert immer noch eine Einbettung des Attraktors, da wir lediglich einen Basiswechsel vorgenommen und somit nur einen Diffeomorphismus angewendet haben.

Die Hauptachsenzerlegung auf verrauschten Daten

Liegt eine Zeitreihe vor, die z.B. aus der Beobachtung eines Experiments hervorgegangen ist, so haben wir bereits in Abschnitt 4.1 erwähnt, dass diese oftmals verrauscht ist. Dies führt dazu, dass – auch bei großem n – keiner der Eigenwerte der Kovarianzmatrix exakt 0 ist. Stattdessen haben diese immer einen gewissen Betrag, da das Rauschen in den meisten Fällen jede Dimension des Raumes betrifft. Da die Zeitreihe selbst verrauscht ist, ist jede Koordinate eines eingebetteten Vektors automatisch von dieser Störung betroffen. Somit würden wir in solchen Fällen

$$n = n'$$

erhalten. Nun kann versucht werden, mit geeigneten Methoden die Stärke des Rauschens herauszufinden, siehe [KS04]. Man nimmt hierbei an, dass das Rauschen additiv auf die Eigenwerte wirkt und für die Eigenwertberechnung lediglich das zeitliche Mittel des Rauschens relevant ist:

$$\sigma_i = \sigma_i^{\text{deterministisch}} + \langle\xi\rangle. \tag{4.17}$$

ξ repräsentiert hierbei den Rauschanteil, von welchem das zeitliche Mittel genommen wird. Haben wir eine realistische Schätzung $\hat{\xi}$ für $\langle\xi\rangle$ so definieren wir nun analog zu (4.15):

$$n' := \#\left\{i \in \{1,\ldots,N\} \mid \sigma_i > \hat{\xi}\right\}. \tag{4.18}$$

Nun können wir mit diesem n' genauso verfahren, wie auf unverrauschten Daten.

4.2.5 Lokale Hauptachsenzerlegung

Die PCA-Methode nimmt lediglich lineare Transformationen auf den Daten vor. Dies führt dazu, dass die PCA nur lineare Strukturen erkennen kann und wir somit keine Möglichkeit haben, die Dimension eines stark gekrümmten Attraktors zu schätzen. Dies war allerdings im obigen Abschnitt auch nicht unser Ziel. Wir wollten direkt einen linearen Unterraum erhalten, der den Attraktor enthält. Um die Dimension des Attraktors mit einem PCA-Verfahren direkt zu schätzen, eignet sich die lokale PCA:
Hierbei nimmt man zunächst ein sogenanntes *Clustering* vor und unterteilt die Menge aller aus der Zeitreihe resultierenden Vektoren im $\mathbb{R}^n$ in K kleinere disjunkte Teilmengen

$$\{\mathbf{x_i} \mid i = 1, \ldots, N\} = \bigcup_{k=1}^{K} X_k \ , \ X_i \cap X_j = \emptyset \ \ \forall i \neq j.$$

Danach führt man auf jedem *Cluster* X_k eine gewöhnliche PCA durch und ermittelt die resultierende Dimension d_k. Zuletzt schätzt man die Dimension des Attraktors A durch

$$\dim(A) \approx \frac{1}{K} \sum_{k=1}^{K} d_k \ . \tag{4.19}$$

Hinter dieser Herangehensweise steht die Hoffnung, dass der Attraktor A selbst eine Mannigfaltigkeit ist. Da der Attraktor dann lokal eine lineare Struktur aufweisen würde, möchten wir versuchen, die Dimension der Mannigfaltigkeit anhand ihrer lokalen Struktur zu schätzen. Voraussetzung für den Erfolg dieser Methode ist, dass die Cluster wirklich lokal sind und Punkte aus einem Cluster eine Abtastung einer linearen Struktur darstellen.
Um die Lokalität der Cluster zu erreichen, können bekannte Clustering-Algorithmen, wie z.B. der *Linde-Buzo-Gray*-Algorithmus [LBG80] verwendet werden.
Eine ausführliche Beschreibung des in dieser Arbeit verwendeten Clustering-Algorithmus kann in Unterabschnitt 6.1.4 gefunden werden. Für eine Motivation dieses Algorithmus sei auf Unterabschnitt 4.2.7 und [BMDG05] verwiesen.

4.2.6 Eignung als Dimensionsschätzer

Wir wollen nun noch auf einige Aspekte hinweisen, die im Kontext der Dimensionsschätzung mit den kennengelernten Schätzern und Dimensionsbegriffen wichtig sind. Auf konkrete Implementierungen und Laufzeitabschätzungen wird in Kapitel 6 näher eingegangen.
Eine Tatsache, die wir bisher immer als selbstverständlich erachtet, jedoch noch nicht explizit erwähnt haben, ist, dass sowohl zum Berechnen der fraktalen Dimensionen als auch in PCA-basierten Verfahren immer bereits eine Einbettung in einen höherdimensionalen Raum vorliegen muss. Wir müssen also eine Einbettung des Attraktors in $\mathbb{R}^n$

vornehmen, bevor wir zur Berechnung der Dimension schreiten können. Hier wird in der Praxis oftmals so verfahren, dass aufsteigend in verschiedene Dimensionen n eingebettet wird. Sobald sich die Schätzung von D_q, bzw. der mit der PCA gefundenen Dimension, nicht weiter mit n erhöht, wird davon ausgegangen, dass sich der Attraktor entfaltet hat und eine Einbettung vorliegt, siehe [KS04, Kru96].
Ein Problem, welches alle Dimensionsschätzer betrifft, ist das Rauschen. Da dieses meistens alle Dimensionen des eingebetteten Raumes betrifft, überschätzen Dimensionsschätzer die eigentlichen Dimensionen häufig. Wir widmen uns diesem Thema lediglich kurz beim PCA-Verfahren. Für eine genauere Auseinandersetzung mit diesem Problem – insbesondere bei fraktalen Dimensionsschätzern – sei auf [KS04] hingewiesen.
Die *Informationsdimension* (Renyi-Dimension der Ordnung 1) eignet sich in der Praxis nicht für verlässliche Schätzungen, weshalb wir diese nicht betrachten.

Der Boxcounting-Dimensionsschätzer

Wie wir bereits bei Definition 4.3 gesehen haben, kommt es bei der Boxcounting-Dimension auf das Verhältnis des Logarithmus der Anzahl okkupierter Würfel zum Logarithmus der Kantenlänge ϵ an, wobei wir den Grenzwert dieses Verhältnisses für ϵ gegen 0

$$d_{cap}(X) = \lim_{\epsilon \searrow 0} \frac{\log Z(\epsilon)}{-\log \epsilon}$$

ermitteln wollen. Beschränkt man sich zunächst auf ein festes ϵ, ist es möglich, ein Kompaktum, welches alle rekonstruierten Punkte enthält, mit Würfeln der Kantenlänge ϵ zu überdecken. Nun muss lediglich die Anzahl der Boxen gezählt werden, welche mindestens einen Punkt beinhalten. Im Anschluss kann man den Quotient der betreffenden Logarithmen ausrechnen.
Wir wollen uns nun den Problemen eines solchen Boxcounting-Schätzers widmen:

- Als erstes muss versucht werden, den Grenzwert $\epsilon \to 0$ zu approximieren. Verkleinert man ϵ immer weiter, um eine Approximation für den Grenzwert zu erhalten, trifft man schnell auf das Problem des endlichen Samplings: Irgendwann ist ϵ so klein, dass disjunkte Punkte in verschiedenen Würfeln liegen und somit genau einem Punkt im Raum genau eine Box zugeordnet ist. Wir nennen das größte ϵ, für welches dieser Fall auftritt, ϵ_0. Es gilt dann

  $$Z(\epsilon) = Z(\epsilon_0) \ \forall \ \epsilon \leq \epsilon_0 \ .$$

  Das bedeutet, dass der Fall

  $$Z(\epsilon) \overset{\epsilon \searrow 0}{\to} Z(\epsilon_0) < \infty$$

  vorliegt. Somit würde d_{cap} gegen 0 gehen. Falls der echte Attraktor keine endliche Punktwolke ist, ist dies offenbar nicht in unserem Sinne. Wir können also ϵ nicht

einfach beliebig klein werden lassen, um die Kapazitätsdimension zu approximieren.

In der Praxis approximiert man den Grenzwert des Quotienten häufig durch die Steigung der Kurve, die entsteht, wenn man den Zähler gegen den Nenner aufträgt. Eine Rechtfertigung hierfür findet man in der Regel von l'Hospital. Wir nehmen an, dass

$$Z(\epsilon) \overset{\epsilon \searrow 0}{\to} \infty$$

gilt. Dies würden wir für den Attraktor erwarten, wenn wir beliebig viele verschiedene Sampling-Punkte hätten. In diesem Fall gehen für ϵ gegen 0 sowohl Zähler als auch Nenner des Boxcounting-Terms aus Gleichung (4.7) gegen unendlich. Für glatten Zähler und Nenner würde die Regel von l'Hospital

$$-\lim_{\epsilon \searrow 0} \frac{\log Z(\epsilon)}{\log \epsilon} = -\lim_{\epsilon \searrow 0} \frac{\frac{\partial \log Z(\epsilon)}{\partial \epsilon}}{\frac{\partial \log \epsilon}{\partial \epsilon}} = -\lim_{\epsilon \searrow 0} \frac{\partial \log Z(\epsilon)}{\partial \log \epsilon}$$

liefern. Es ist anzumerken, dass die Funktion Z im Zähler unstetig ist und diese Approximation somit nur eine Heuristik ist, da wir die Regel von l'Hospital nicht anwenden können.

In der Praxis bildet man nun finite Differenzen

$$-\lim_{\epsilon \searrow 0} \frac{\partial \log Z(\epsilon)}{\partial \log \epsilon} \approx -\frac{\log Z(\epsilon_1) - \log Z(\epsilon_2)}{\log \epsilon_1 - \log \epsilon_2}$$

für kleine Parameter $\epsilon_1 > \epsilon_2 > \min \{||\mathbf{x_i} - \mathbf{x_j}||_\infty \mid \mathbf{x_i} \neq \mathbf{x_j};\ i, j \in \{1, \ldots, N\}\}$.

Eine Betrachtung des Approximationsfehlers ist hier leider nicht ohne Weiteres möglich, da wir weder den Fehler, den ein unglattes Z liefert, noch den Fehler, den die Vernachlässigung des Grenzwertes für ϵ gegen 0 liefert, kennen. Beide hängen stark von der Struktur des Attraktors sowie der Qualität der Zeitreihe ab. Für eine nähere Betrachtung dieses Problems sei auf [Hun90] verwiesen.

- Ein weiteres Problem, welches hier auftritt, ist die Anzahl der rekonstruierten Punkte, die benötigt werden, um für ein festes ϵ eine gute Schätzung für den Grenzwert gegen 0 zu erhalten. Wir haben bereits gesehen, dass eine endliche Anzahl von Punkten generell zu einem Problem beim Grenzübergang führt, doch in [ER92] wird zusätzlich angemerkt, dass man für verlässliche Dimensionsschätzungen mit endlich vielen rekonstruierten Punkten sehr viele Daten benötigt. Dort wird gezeigt, dass die Anzahl der benötigten Daten von der Dimension des Attraktors abhängt. Für eine verlässliche Schätzung wird

$$d_{cap} < 2 \log_{10} N \tag{4.20}$$

gefordert. Eine Erklärkung und Herleitung dieser Ungleichung findet sich auch in [KS04]. Generell ist dies ein Problem, was bei allen fraktalen Dimensionsschätzern

auftritt, siehe [Cam03].

Sind wir jedoch nicht an der exakten Attraktordimension interessiert und genügt uns eine hinreichend große Einbettungsdimension, so besagt eine Anmerkung in [Eng91], dass wir eventuell auch mit weniger Datenpunkten auskommen.

- Ein Nachteil, welcher ausschließlich den Boxcounting-Schätzer betrifft, ist die Tatsache, dass dieser nicht auf das dem Attraktor zugrundeliegende Maß μ eingeht. Dass dies problematisch sein kann, haben wir bereits im Unterabschnitt 4.2.2 erwähnt. Der Boxcounting-Schätzer unterscheidet lediglich zwischen okkupierten und leeren Boxen. Die Anzahl der Punkte in einer okkupierten Box ist irrelevant. Dies kann bei einem Attraktor A mit Maß μ, das stark vom uniformen Maß abweicht, zu einem Problem werden [GWST82]. Bei einer endlichen Anzahl von Samples kann es vorkommen, dass sehr viele Punkte benötigt werden, damit eine Box B, für welche das Maß
$$1 \gg \mu(A \cap B) > 0 \tag{4.21}$$
sehr klein ist, belegt wird. Ist diese Box allerdings okkupiert, so bekommt sie im Boxcounting-Schätzer dieselbe Gewichtung wie eine Box, welche ein großes Maß hat. Bei einem kontinuierlichen Attraktor würde dieses Problem im Grenzwert für ϵ gegen 0 verschwinden, da wir auch Gebiete mit großem Maß beliebig gut zerlegen könnten. Wir haben allerdings oben beschrieben, warum wir bei einer endlichen Datenmenge ϵ nicht beliebig klein werden lassen können.

Der Korrelationsdimensionsschätzer

Um die Korrelationsdimension zu schätzen benutzen wir Formel (4.9)

$$d_{cor} := \lim_{\epsilon \searrow 0} \lim_{N \to \infty} \frac{\log \overbrace{\left(\frac{2}{N(N-1)} \cdot \sum_{i=1}^{N} \sum_{j=i+1}^{N} H\left(\epsilon - ||\mathbf{x_i} - \mathbf{x_j}||_2\right) \right)}^{C(\epsilon) :=}}{\log \epsilon}.$$

Den Grenzwert für N gegen unendlich vernachlässigen wir, da wir ohnehin eine feste Anzahl an Samples vorliegen haben. Wäre ϵ fest, so müssten wir also lediglich alle ϵ-Paare finden. Ein ϵ-Paar sei hierbei definiert durch

$$(\mathbf{x_i}, \mathbf{x_j}) \in \mathbb{R}^n \times \mathbb{R}^n \text{ ist ein } \epsilon\text{-Paar} \Leftrightarrow ||\mathbf{x_i} - \mathbf{x_j}||_2 \leq \epsilon \text{ und } i \neq j.$$

Nun wollen wir die Probleme eines Korrelationsdimensionsschätzers betrachten:

- Zunächst ist der Grenzübergang für ϵ gegen 0 zu bilden. Wir können diesen jedoch auch hier aufgrund der endlichen Datenmenge nicht vollziehen, da wir ab einem gewissen kleinen ϵ_0 immer dieselben ϵ-Paare finden für $0 < \epsilon < \epsilon_0$. Dies sind genau

die 2-Tupel

$$\{(\mathbf{x_i}, \mathbf{x_j}) \mid \mathbf{x_i} = \mathbf{x_j},\ i \neq j\}\,.$$

Dies ist offensichtlich nicht in unserem Sinne, da wir wieder einen Attraktor annehmen, der keine endliche Punktwolke ist.

Hier wird – durch dieselbe Heuristik wie beim Boxcounting-Dimensionsschätzer – in der Praxis durch

$$\lim_{\epsilon \to 0} \frac{\partial \log C(\epsilon)}{\partial \log \epsilon} \approx \frac{\log C(\epsilon_1) - \log C(\epsilon_2)}{\log \epsilon_1 - \log \epsilon_2}$$

für kleine Parameter $\epsilon_1 > \epsilon_2 > \min\{||\mathbf{x_i} - \mathbf{x_j}||_2 \mid \mathbf{x_i} \neq \mathbf{x_j};\ i, j \in \{1, \ldots, N\}\}$ approximiert. Doch auch hier ist dies lediglich eine Heuristik, da wir die Regel von l'Hospital nicht auf den unglatten Zähler anwenden können.

- Für den Korrelationsschätzer gelten dieselben Restriktionen bezüglich der Anzahl der benötigten Punkte, um eine gute Schätzung zu erhalten, wie beim Boxcounting-Schätzer in Gleichung (4.20).

- Ein weiteres Problem, welches in der Praxis beim Schätzen der Korrelationsdimension auftritt, ist die Tatsache, dass wir ungewollte zeitliche Korrelationen vorliegen haben. Der Hintergrund dieser Tatsache ist, dass wir davon ausgehen, dass die vorliegenden rekonstruierten Vektoren ein – nach der Verteilung auf dem Attraktor gezogenes – zufälliges Sampling darstellen. Dieses wollen wir nutzen, um räumliche Korrelationen zu messen und damit die Korrelationsdimension des Attraktors zu bestimmen. Jedoch kommt es für ein festes ϵ und einen kleinen Zeitparameter τ vor, dass wir Vektoren $\mathbf{x_i}$ erhalten, für welche

  $$\exists\ K \in \mathbb{N} \setminus \{0\} : (\mathbf{x_i}, \mathbf{x_{i+k}}) \text{ ist ein } \epsilon\text{-Paar } \forall\ k = 1, \ldots, K$$

  gilt.

  Somit sind die Vektoren $\mathbf{x_i}, \ldots, \mathbf{x_{i+k}}$ zeitlich korreliert und unser Schätzer trägt einen *Bias*, da wir davon ausgehen wollten, dass unsere Samples unabhängig voneinander gezogen wurden.

Um das letztgenannte Problem der zeitlichen Korrelation zu umgehen, schlug James Theiler eine denkbar einfache Korrektur des Schätzers vor [The86]:

$$C(\epsilon) := C(\epsilon, n_{\min}) = \left(\frac{2}{(N - n_{\min})(N - n_{\min} - 1)} \cdot \sum_{i=1}^{N} \sum_{j=i+1+n_{\min}}^{N} H\left(\epsilon - ||\mathbf{x_i} - \mathbf{x_j}||_2\right) \right) \tag{4.22}$$

Wir messen also nur noch die räumliche Korrelation zwischen Samples, die mindestens $n_{\min} + 1$ Zeitschritte auseinander liegen.

Wir wollen nun unsere Definition eines ϵ-Paares entsprechend abändern:

$$(\mathbf{x_i}, \mathbf{x_j}) \in \mathbb{R}^n \times \mathbb{R}^n \text{ ist ein } \epsilon\text{-Paar} \Leftrightarrow ||\mathbf{x_i} - \mathbf{x_j}||_2 \leq \epsilon \text{ und } |i-j| > n_{\min} + 1.$$

Neben den in Abschnitt 4.3 beschriebenen Methoden, eignet sich eine in [PSVM92] vorgeschlagene Methode besonders für das Schätzen von $n_{\min}$. Diese wurde im Rahmen dieser Arbeit zwar nicht implementiert, soll aber trotzdem kurz dargestellt werden:
$n_{\min}$ wird durch einen sogenannten *Raum-Zeit-Seperationsplot* ermittelt. Das Ziel ist es, die minimale Zeitspanne zu finden, die vergehen muss, damit keine zusätzlichen ϵ-Paare mehr gefunden werden, die auf zeitliche Korrelation zurückzuführen sind:
Die Idee hierbei ist die Anzahl der ϵ-Paare abhängig von den Variablen $n_{\min}$ und ϵ zu betrachten. Wir bezeichnen diese Anzahl im Folgenden mit

$$N_{\text{near}}(\epsilon, n_{\min}) \ .$$

Wir betrachten einen fixen Anteil aller möglichen Paare

$$k(n_{\min}) := a \cdot N_{\text{near}}(\infty, n_{\min}), a \in (0,1) \tag{4.23}$$

und suchen

$$\epsilon_0(n_{\min}) := \min \{\epsilon \mid N_{\text{near}}(\epsilon, n_{\min}) \geq k(n_{\min})\} \, .$$

Im Anschluss wählen wir n_{sat} als die kleinste natürliche Zahl, für welche

$$\epsilon_0(n_{\text{sat}}) \geq \epsilon_0(n_{\min}) \ \forall N \gg c \geq n_{\min} \geq n_{\text{sat}} \tag{4.24}$$

gilt.
Das maximal zu betrachtende $n_{\min}$ sollte hierbei deutlich kleiner als die Gesamtzahl N aller Vektoren gewählt werden, weil wir sonst nur noch sehr wenige Paare betrachten können, da $N_{\text{near}}(\infty, n_{\min})$ dann sehr klein wird und unsere Betrachtungen nicht mehr sinnvoll sind. Deshalb sollten wir eine feste obere Schranke c angeben.
Eine gute Wahl für c zu treffen ist schwierig, da die Größe einer sinnvollen oberen Schranke maßgeblich von der zeitlichen Korrelation der Zeitreihe abhängt.
Erhalten wir $n_{\text{sat}} = c$, kann dies auf die ungenügende Länge der Zeitreihe zurückzuführen sein, siehe [KS04]. Dies können wir so interpretieren, dass sämtliche Datenpunkte zeitlich korreliert sind.
In der Praxis wählt man verschiedene äquidistante $a_i \in (0,1)$ als a in Gleichung (4.23) und verwendet schlussendlich

$$\max_i n_{\text{sat}}(a_i) \tag{4.25}$$

als $n_{\min}$ in der Theiler-Korrektur (4.22).

Die Hauptachsenzerlegung

Wie wollen nun die Eigenschaften und Probleme eines PCA-Schätzers erörtern:

- Es ist sofort klar, dass wir auf verrauschten Daten Probleme bekommen, sobald das Level des Rauschens größer wird als die unverrauschten Eigenwerte. In diesem Fall würden wir Eigenvektoren bei der Transformation der Trajektorie vernachlässigen, obwohl sie einen relevanten Anteil zum Span von T beitragen. Des Weiteren muss das Rauschen wirklich in der additiven Gestalt von (4.17) vorliegen und selbst dann müssen wir uns darauf verlassen können, dass unser Schätzer für $\hat{\xi}$ korrekte Ergebnisse liefert. Wir können jedoch auf verrauschten Daten nicht mehr garantieren, dass wir nach Anwendung der PCA wirklich eine Einbettung des Attraktors erhalten.

  Es gibt eine weitere Möglichkeit mit verrauschten Daten umzugehen, indem man alle Eigenwerte vernachlässigt, die prozentual – gemessen am größten Eigenwert – einen geringen Betrag aufweisen. In diesem Fall kann vermutet werden, dass dieser Betrag auf eine Störung zurückzuführen ist. Hierzu nehmen wir wieder an, dass die Eigenwerte absteigend sortiert sind. Wir setzen

  $$n' := \max\left\{ i \in \{1, \dots, N\} \mid \frac{\sigma_i}{\sigma_1} > c \right\}, \tag{4.26}$$

  wobei $c \in (0, 1)$ der Prozentsatz ist, ab welchem wir einen Eigenwert als relevant betrachten. Jedoch kann auch mit dieser Methode nicht garantiert werden, dass wir ausschließlich solche Eigenwerte vernachlässigen, die aufgrund des Rauschens nicht Null sind.

- Des Weiteren ist anzumerken, dass die PCA ein lineares Dimensionsreduktionsverfahren ist. Dies bedeutet, dass wir lediglich lineare Strukturen erkennen können. In unserem Fall ist dies jedoch anscheinend kein Nachteil, denn es ist nicht der Attraktor selbst, den wir mit der PCA finden wollen, sondern lediglich der kleinstmögliche **lineare Unterraum** des $\mathbb{R}^n$, welcher den Attraktor enthält. Dies ist auch der Unterschied zu den Dimensionsschätzern, die sich am Renyi-Dimensionsbegriff orientieren: Bei diesen versuchen wir die (Boxcounting)-Dimension m des Attraktors möglichst gut zu schätzen und dann den Prozess in $\mathbb{R}^{\lceil 2m+1 \rceil}$ einzubetten, wohingegen wir hier anhand einer hochdimensionalen Einbettung versuchen, den kleinsten linearen Unterraum zu finden, in welchen der Prozess eingebettet ist. Im optimalen Fall ist dies direkt ein Unterraum der Dimension kleiner oder gleich $\lceil 2m + 1 \rceil$.

  Ein Problem ergibt sich durch die Linearität der PCA dennoch, vor allem bei großem n, wenn sich mit steigender Einbettungsdimension die Struktur des abgetasteten Attraktors im $\mathbb{R}^n$ so ändert, dass dieser sich in alle Raumrichtungen ausdehnt. In diesem Fall liegt (fast sicher) eine Einbettung vor, jedoch haben wir keine Möglichkeit mehr die PCA erfolgreich anzuwenden. Zum einen kann dieses

Problem auftreten, wenn sich die nächstgrößere Mannigfaltigkeit, welche den Attraktor enthält, stark krümmt und sich dadurch so durch alle Raumdimensionen legt, dass ein Anwenden der PCA keinen kleineren linearen Raum findet, der die Mannigfaltigkeit enthält. Hier sehen wir, dass die Linearität der PCA dann doch eine starke Einschränkung ist und es uns oftmals nicht möglich macht, eine zuverlässige Schätzung für eine möglichst kleine Einbettungsdimension zu erhalten. Zum anderen tritt dieses Problem in der Praxis oft dadurch auf, dass der Zeitparameter τ (vgl. Theorem 3.3) zu groß gewählt wurde und die Vektoren in $\mathbb{R}^n$ absolut unkorreliert scheinen, siehe [KS04, BK85]. Dies kann behoben werden, indem man die Zeitskala bei der Beobachtung eines Experimentes feiner auflöst. In manchen Fällen ist dies jedoch nicht möglich und man hat keine Chance mit der PCA ein sinnvolles Ergebnis zu erzielen. Ein Beispiel für letzteren Fall ist eine diskrete Zeitreihe, bei der wir keine Möglichkeit haben, die Sampling-Rate weiter zu erhöhen. Näheres zur Wahl von τ folgt in Abschnitt 4.3.

- Wir ahmen die Delay-Einbettung lediglich bis auf Diffeomorphie genau nach. Es kann möglich sein, dass eine direkte Delay-Einbettung in $\mathbb{R}^{\lceil 2m+1 \rceil}$ bessere numerische Ergebnisse liefert als die Herangehensweise über die PCA.. Für eine genauere Betrachtung dieses Problems sei [KS04] empfohlen.

- Obwohl wir die Laufzeitbetrachtungen erst in Kapitel 6 näher ausführen wollen, sei hier bereits darauf hingewiesen, dass die Laufzeit der PCA nicht exponentiell in der Raumdimension n skaliert. Somit ist es uns hier möglich, auch große n als erste Einbettungsdimension zu betrachten, ohne wesentlich längere Laufzeiten in Kauf zu nehmen. Wir können also, entgegen der oben vorgeschlagenen Methode des Inkrementierens von n, auch ein so großes n wählen, dass wir sicher sein können, direkt eine Einbettung des Attraktors vorliegen zu haben, da die Dimensionen der meisten Attraktoren, die man in der Praxis untersucht, klein sind. Einer beliebig großen Dimension n steht jedoch die implizit gemachte Forderung entgegen, dass die Länge unserer Zeitreihe immer wesentlich größer sein sollte als die Einbettungsdimension, da wir sonst nur wenige hochdimensionale Punkte erhalten und somit den Attraktor nur schlecht abgetastet haben.

  Wir wollen hier nicht genauer auf diese Probleme eingehen, jedoch sei darauf hingewiesen, dass eine gute Wahl von n und τ wichtig ist. Das Problem liegt meistens darin, eine gute Kombination beider Parameter zu finden. Eine Methode zur Schätzung von $n \cdot \tau$ finden wir in [BK85]. Dort wird vorgeschlagen, durch Betrachten des Fourierspektrums der Zeitreihe einen guten Wert für diese Parameter zu finden. Mit der Schätzung von τ – unabhängig von n – beschäftigen wir uns ausführlicher in Abschnitt 4.3.

Die lokale Hauptachsenzerlegung

Die lokale Hauptachsenzerlegung eignet sich – im Gegensatz zur globalen Variante – nicht zum direkten Finden des Raumes, in welchem sich der Attraktor bereits entfaltet hat. Dies liegt daran, dass wir zwar einzelne lokale Schätzungen für diesen Raum erhalten, jedoch können wir keine direkte, globale Transformation der Daten – wie in Gleichung (4.16) – angeben. Im Allgemeinen sind die lokalen Transformationen nicht stetig zusammensetzbar, weshalb wir uns also mit einer Schätzung für die Attraktordimension begnügen müssen.
Eine Voraussetzung, damit auch wirklich die Dimension des Attraktors approximiert werden kann, ist, dass sich der Attraktor lokal linear verhält. Hat dieser also Mannigfaltigkeitenstruktur, so ist eine gute Schätzung möglich. Bei allgemeinen fraktalen Attraktoren ist dies jedoch nicht der Fall.

4.2.7 Hochdimensionale Räume und Abstände

Findet die Einbettung der Zeitreihe in $\mathbb{R}^d$ statt, kann es für große d zu unintuitivem Verhalten der euklidischen Norm kommen. Zu welchen Problemen dies führen kann, soll hier näher erläutert werden.

Concentration of Norms

Als Concentration of Norms-, bzw. allgemeiner *Concentration of Measure (CoM)*-Effekt wird ein Konzentrationseffekt bezeichnet, welcher sich vor allem auf die euklidische Norm in hochdimensionalen Räumen auswirkt. In [LV07] kann in diesem Zusammenhang ein wichtiger Satz von Demartines gefunden werden.

Satz 4.6 [Concentration of Norms]
Sei $\mathbf{y} = (y_1, \ldots, y_d)$ *ein d-dimensionaler Vektor, dessen Koordinaten unabhängig voneinander, nach derselben Verteilung gezogen wurden. Sei außerdem* $\mathbb{E}\left[(y_i - \mathbb{E}[y_i])^8\right] < \infty$. *Dann gilt*

$$\begin{aligned} \rho := \mathbb{E}[||\mathbf{y}||_2] &= \sqrt{ad - b} + O\left(\frac{1}{d}\right), \\ \sigma^2 := Var(||\mathbf{y}||_2) &= b + O\left(\frac{1}{\sqrt{d}}\right), \end{aligned}$$

wobei $a, b \in \mathbb{R}$ *von d unabhängig sind.*

Dies bedeutet, dass mit wachsender Dimension d der Erwartungswert der euklidischen Norm eines Zufallsvektors mit $O(\sqrt{d})$ ansteigt, während die Varianz für große d nahezu konstant ist. Nach dem Satz von Tschebyscheff gilt unter den Voraussetzungen des

Satzes 4.6

$$\mathbb{P}\left(\mid ||\mathbf{y}||_2 - \rho \mid \geq \epsilon\right) \leq \frac{b + O\left(\frac{1}{\sqrt{d}}\right)}{\epsilon^2} \xrightarrow{d\to\infty} \frac{b}{\epsilon^2}.$$

Dies verdeutlicht, dass die Norm eines Zufallsvektors in hohen Dimensionen größer wird. Hierzu sei auch auf Abbildung 4.1 hingewiesen.

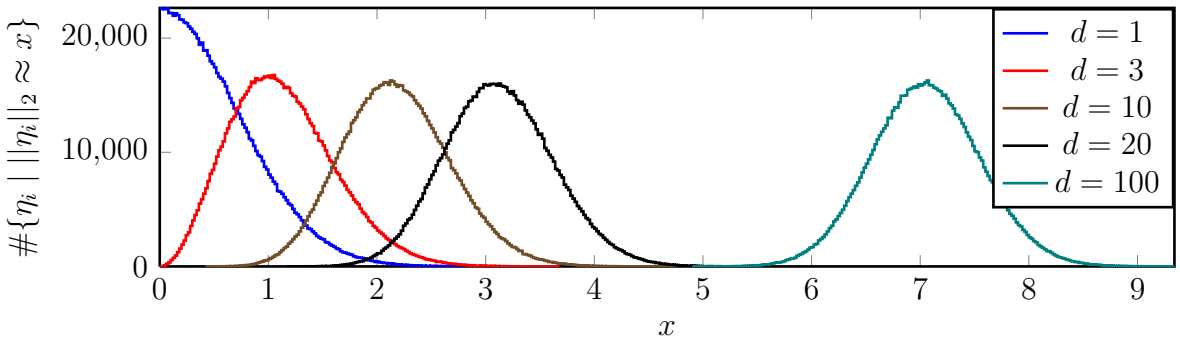


Abb. 4.1: *Bin*-Plot für die Norm von 10^6 $N(\mathbf{0}, \mathbf{I})$-Zufallsvektoren $\eta_i \in \mathbb{R}^d$, $i = 1, \ldots, 10^6$ in verschiedenen Dimensionen d

Besonders illustrativ ist dieser Effekt auch in [Mur09] dargestellt, wobei der dort verwendete Begriff der *Ultrametricity* eng mit dem Concentration of Measure-Effekt verwandt ist.
Das CoM-Phänomen führt zu Problemen, bei der Messung von Abständen. Sind die vorliegenden Daten in hohen Dimensionen zufällig verteilt so ergibt sich beispielsweise ein Problem bei der Nachbarsuche. Man stellt fest, dass der Abstand zweier Vektoren für alle möglichen Vektorpaare annähernd gleich ist. Dies ist darauf zurückzuführen, dass auch die Differenz zweier Vektoren wieder ein Zufallsvektor ist. Neben der Tatsache, dass die euklidische Norm somit kein geeignetes Abstandsmaß zwischen Punkten mehr liefert, führt dies zu numerischer Instabilität.
Wir gehen zwar nicht davon aus, dass unsere eingebetteten Vektoren zufällig sind, aber in der Praxis spielen verrauschte Daten eine Rolle und auf diesen führt der oben genannte Effekt zum beschriebenen Problem.
Konkret bieten sich uns an zwei Stellen Möglichkeiten, dieses Problem zu umgehen:

- Im Korrelationsdimensionsschätzer kann die euklidische Norm im Argument der Heavyside-Funktion durch einen anderen Distanzbegriff ersetzt werden:

$$H(\epsilon - ||\mathbf{x} - \mathbf{y}||_2) \to H(\epsilon - d(\mathbf{x}, \mathbf{y})).$$

- Im Clustering-Algorithmus, der als Vorverarbeitungsschritt der lokalen PCA dient, kann ein anderer Distanzbegriff bei der Suche geeigneter Clustermittelpunkte verwendet werden. Näheres zum Clustering-Algorithmus folgt im Unterabschnitt 6.1.4.

In [VF05] wird erwähnt, dass für den Abstand

$$\overline{\mathcal{D}} := \left| \max_{\mathbf{x}} ||\mathbf{x}||_p - \min_{\mathbf{x}} ||\mathbf{x}||_p \right|$$

zwischen der größten und der kleinsten l_p-Norm in einem Datensatz aus Zufallsvektoren folgendes zu beobachten ist:

- Für $p < 2$ gilt: $\overline{\mathcal{D}} \overset{d\to\infty}{\longrightarrow} \infty$.
- Für $p = 2$ gilt: $\overline{\mathcal{D}} \overset{d\to\infty}{\longrightarrow}$ const.
- Für $p > 2$ gilt: $\overline{\mathcal{D}} \overset{d\to\infty}{\longrightarrow} 0$.

Dabei ist $||\cdot||_p$ die *Minkowski-p-Norm* für $p \in \mathbb{Q}^+ \setminus \{0\}$:

$$||(x_1, \ldots, x_d)^T||_p := \left(\sum_{i=1}^{d} |x_i|^p \right)^{\frac{1}{p}}.$$

Für eine genauere Untersuchung dieses Verhaltens von $\mathcal{D}$ sei insbesondere auf [AHK01] hingewiesen.
In hohen Dimensionen ist somit die Aufgabe des Clusterings, bzw. der Nachbar-Suche mit l_p-Normen mit $p > 2$ sinnlos, da die Norm kein geeignetes Abstandsmaß mehr liefert. In [AHK01] wird gezeigt, dass es sinnvoller ist die l_1- oder $l_{\frac{1}{2}}$-Norm zu verwenden und dass dies auch zu besseren numerischen Ergebnissen in Clustering-Problemen führt als die l_2-Norm. Es sei angemerkt, dass die l_2-Norm für normalverteiltes Rauschen allerdings stabilere Ergebnisse liefert als Normen mit kleinerem p, siehe [VF05].

Bregman-Divergenzen und Clustering

Neben den Minkowski-Normen sollen auch allgemeine *Bregman-Divergenzen* als Distanzmaß motiviert werden. Divergenzen sind im Allgemeinen unsymmetrische Abstandsmaße und es ist zunächst fraglich, ob deren Verwendung in unserem Kontext sinnvoll ist. Wir wollen in diesem Unterabschnitt den Einsatz von Bregman-Divergenzen im Clustering-Algortihmus motivieren. Wir orientieren uns im Folgenden an [BMDG05].
Neben den in der Wahrscheinlichkeitstheorie häufig verwendeten *Csiszár-Divergenzen*, finden Bregman-Divergenzen mittlerweile große Bedeutung als Diskrepanzmaß. Praktisch können auch verschiedene Csiszár-Divergenzen als Abstand zwischen Punkten verwendet werden, jedoch ist die folgende Theorie und mit ihr die Verwendung der Divergenzen als Abstandsmaß zwischen Punkten im $\mathbb{R}^n$ nur für Bregman-Divergenzen gerechtfertigt, weshalb wir auf die explizite Betrachtung von Csiszár-Divergenzen verzichten.

Definition 4.7 [Bregman-Divergenz]
Sei $S \subset \mathbb{R}^d$ konvex und $\phi : S \to \mathbb{R}$ im Inneren $\mathring{S}$ von S differenzierbar. Die ***Bregman-***

Divergenz $d_\phi : S \times \mathring{S} \to \mathbb{R}^+$ *ist definiert als*

$$d_\phi(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x}) - \phi(\mathbf{y}) - \langle \mathbf{x} - \mathbf{y}, \nabla\phi(\mathbf{y}) \rangle \,. \tag{4.27}$$

Eine Bregman-Divergenz erfüllt zwar die Bedingungen

- $d_\phi(x, y) \geq 0$ und
- $d_\phi(x, y) = 0 \Leftrightarrow x = y$,

allerdings ist d_ϕ im Allgemeinen nicht symmetrisch und die Dreiecksungleichung muss nicht gelten. Dies führt dazu, dass Bregman-Divergenzen im Allgemeinen keine Metriken sind. Dass diese dennoch für Abstandsmessungen im Clustering-Algorithmus interessant sind, folgt aus der engen Beziehung zwischen Bregman-Divergenzen und exponentiellen Familien von Wahrscheinlichkeitsmaßen. Eine explizite Herleitung dieser Zusammenhänge soll hier nicht erfolgen. Es sollen lediglich die Ergebnisse präsentiert werden. Für eine ausführliche Behandlung dieses Themas und detaillierte Beweise sei auf [BMDG05] verwiesen.

Lemma 4.8
Sei $(\Omega, \mathcal{A}, \mathbb{P})$ *ein Wahrscheinlichkeitsraum und* $X : \Omega \to \mathring{S}$ *eine nach dem Wahrscheinlichkeitsmaß* ξ *verteilte Zufallsvariable, wobei* S *wie in Definition 4.7 sei. Dann gilt für eine beliebige Bregman-Divergenz* d_ϕ

$$\mathbb{E}_\xi[X] = \arg\min_{c \in \mathring{S}} \mathbb{E}_\xi \left[d_\phi(X, c) \right] .$$

Dieses Lemma sagt aus, dass der Erwartungswert einer Zufallsvariable X im Sinne jeder Bregman-Divergenz die beste konstante Vorhersage für X liefert. Dies ist von entscheidender Bedeutung für den Clustering-Algorithmus, der in Unterabschnitt 6.1.4 vorgestellt wird, da wir diese Relation dort nutzen werden, um in jedem Iterationsschritt die neuen Clustermittelpunkte zu finden.

Definition 4.9 [Exponentielle Familie]
Sei $\theta \in \mathbb{R}^d$ *beliebig und* p_0 *eine beliebige Wahrscheinlichkeitsdichte auf einer konvexen Menge* $S \subset \mathbb{R}^d$ *mit der Borel-Algebra versehen, sodass*

$$\exp\left(\Psi(\theta)\right) := \int_S \exp\left(\theta^T \mathbf{x}\right) \cdot p_0(\mathbf{x}) d\mathbf{x} < \infty$$

ist.[2] *Dann nennen wir die Familie*

$$P_{\theta,\Psi}(\mathbf{x}) := \exp\left(\theta^T \mathbf{x} - \Psi(\theta)\right) \cdot p_0(\mathbf{x}) \tag{4.28}$$

[2] Ψ nennt man *Kumulantenfunktion.*

von Wahrscheinlichkeitsdichten eine ***exponentielle Familie*** *in den Parametern* θ *und* Ψ*.*

Der entscheidende Zusammenhang zwischen Bregman-Divergenzen und exponentiellen Familien ist nun durch das folgende Theorem gegeben:

Theorem 4.10
Zu jeder exponentiellen Familie von Wahrscheinlichkeitsdichten $P_{\theta,\Psi}(\mathbf{x})$ *auf einer konvexen Menge* S *gibt es genau eine konvexe Funktion* $\phi_\Psi : S \times \mathring{S} \to \mathbb{R}^+$*, welche die Darstellung*

$$P_{\theta,\Psi}(\mathbf{x}) = \exp\left(-d_{\phi_\Psi}(\mathbf{x}, \mu(\theta))\right) b_{\phi_\Psi}(\mathbf{x}) \tag{4.29}$$

erlaubt. $\mu(\theta)$ *ist hierbei der Erwartungswert einer nach* $P_{\theta,\Psi}$ *verteilten Zufallsvariablen und* b_{ϕ_Ψ} *eine Funktion, die durch* ϕ_Ψ *eindeutig bestimmt ist.*

Dieser Zusammenhang zeigt die Relevanz der Bregman-Divergenzen beim Clustering. Um dies zu verdeutlichen, nehmen wir nun an, dass N Datenpunkte $\mathbf{x_i} \in \mathbb{R}^d$, $i = 1, \ldots, N$ und M Clustermittelpunkte $\mathbf{c_i} \in \mathbb{R}^d$, $i = 1, \ldots, M$ vorliegen und bekannt ist, dass die Datenpunkte mittels der zu $P_{\theta,\Psi}$ gehörenden Verteilung verrauscht wurden. Bevor im Clustering-Algorithmus das Lemma 4.8 zum Berechnen der neuen Clustermittelpunkte verwendet werden kann, muss jeder Datenpunkt einem Cluster zugeordnet werden. Um den zu $\mathbf{x_i}$ korrespondierenden Clustermittelpunkt zu finden, wird eine *Maximum-Likelihood*-Analyse durchgeführt und der Clustermittelpunkt $\mathbf{c}(\mathbf{x_i})$ ermittelt, welchem $\mathbf{x_i}$ unter der Annahme der Verteilungsdichte $P_{\theta,\Psi}$ am wahrscheinlichsten zugeordnet ist:

$$\begin{aligned} c(\mathbf{x_i}) &:= \arg\max_{j=1,\ldots,M} \exp\left(-d_{\phi_\Psi}(\mathbf{x_i}, \mathbf{c_j})\right) b_{\phi_\Psi}(\mathbf{x_i}) && (4.30)\\ &\overset{(*)}{=} \arg\max_{j=1,\ldots,M} \exp\left(-d_{\phi_\Psi}(\mathbf{x_i}, \mathbf{c_j})\right) && (4.31)\\ &= \arg\min_{j=1,\ldots,M} d_{\phi_\Psi}(\mathbf{x_i}, \mathbf{c_j}), && (4.32) \end{aligned}$$

wobei $(*)$ darauf zurückzuführen ist, dass $b_{\phi_\Psi}(\mathbf{x})$ nicht von den $\mathbf{c_j}$ abhängt.
Dies bedeutet, dass bei einer Störung der Daten mittels einer exponentiellen Familie das Problem der Clusterzuweisung auf die korrespondierende Bregman-Divergenz übertragen werden kann.
Im Clustering-Algorithmus wird eine Bregman-Divergenz vorgegeben, mit welcher die Berechnungen durchgeführt werden. Hat man eine Vermutung mit welcher Verteilung die Daten verrauscht wurden, kann man diese Vorinformation nutzen, indem man beim Clustering die entsprechende Bregman-Divergenz verwendet.
Zuletzt soll der den Zusammenhang zwischen Bregman-Divergenzen und exponentiellen Familien anhand eines Beispiels veranschaulicht werden:

Beispiel 4.1: [Normalverteilung] Die Normalverteilung $N(\nu, \sigma^2 \cdot \mathbf{I})$ mit fester Kovarianzmatrix $\sigma^2 \cdot \mathbf{I}$ stellt eine exponentielle Familie bezüglich des Mittelwertes ν dar:

Die Wahl $\theta = \frac{\nu}{\sigma^2}$ und $p_0(\mathbf{x}) = \frac{1}{\sqrt{(2\pi\sigma^2)^d}} \exp\left(-\frac{\mathbf{x}^T\mathbf{x}}{2\sigma^2}\right)$ führt zur Kumulantenfunktion

$$\begin{aligned}\Psi(\theta) &= \log\left(\frac{1}{\sqrt{(2\pi\sigma^2)^d}} \int_{\mathbb{R}^d} \exp\left(\frac{2\mathbf{x}^T\nu - \mathbf{x}^T\mathbf{x}}{2\sigma^2}\right) d\mathbf{x}\right) \\ &= \log\left(\frac{1}{\sqrt{(2\pi\sigma^2)^d}} \int_{\mathbb{R}^d} \exp\left(\frac{-||\mathbf{x}-\nu||_2^2 + \nu^T\nu}{2\sigma^2}\right) d\mathbf{x}\right) \\ &= \frac{\nu^T\nu}{2\sigma^2} = \frac{\sigma^2}{2}\theta^T\theta.\end{aligned}$$

Mit diesem Ψ erhalten wir die exponentielle Familie

$$\begin{aligned}P_{\theta,\Psi}(x) &:= \frac{1}{\sqrt{(2\pi\sigma^2)^d}} \exp\left(\mathbf{x}^T\frac{\nu}{\sigma^2} - \frac{\sigma^2}{2}\frac{\nu^T\nu}{\sigma^2\cdot\sigma^2}\right) \exp\left(-\frac{\mathbf{x}^T\mathbf{x}}{2\sigma^2}\right) \\ &= \frac{1}{\sqrt{(2\pi\sigma^2)^d}} \exp\left(-\frac{1}{2\sigma^2}\cdot||\mathbf{x}-\nu||_2^2\right)\end{aligned}$$

und somit die gewünschte Normalverteilung.
Wie man sieht ist die zur Normalverteilung korrespondierende Bregman-Divergenz durch die auf $\mathbb{R}^d \times \mathbb{R}^d$ definierte quadrierte euklidische Norm der Vektorendifferenz gegeben. Diese ergibt sich für $\phi(\mathbf{z}) = \frac{1}{2\sigma^2}\mathbf{z}^T\mathbf{z}$:

$$\begin{aligned}d_\phi(\mathbf{x},\mathbf{y}) &= \frac{1}{2\sigma^2}\left(\mathbf{x}^T\mathbf{x} - \mathbf{y}^T\mathbf{y} - 2\langle\mathbf{x}-\mathbf{y},\mathbf{y}\rangle\right) \\ &= \frac{1}{2\sigma^2}\left(\mathbf{x}^T\mathbf{x} + \mathbf{y}^T\mathbf{y} - 2\langle\mathbf{x},\mathbf{y}\rangle\right) \\ &= \frac{1}{2\sigma^2}||\mathbf{x}-\mathbf{y}||_2^2.\end{aligned}$$

Eindimensionale exponentielle Familien

Einige Beispiele eindimensionaler exponentieller Familien und der korrespondierenden Bregman-Divergenzen sind in Tabelle 4.2 dargestellt.
Da wir davon ausgehen, dass jedes Element der Zeitreihe mit derselben Verteilung gestört wurde und dies unabhängig voneinander geschieht, ergibt sich im d-dimensionalen Fall die entsprechende Bregman-Divergenz als Summe der eindimensionalen Bregman-Divergenzen. Analog zu (4.30) gilt

$$c(\mathbf{x_i}) := \arg\max_{j=1,\ldots,M} \prod_{k=1}^{d} \exp\left(-d_{\phi_\Psi}(x_i^k, c_j^k)\right) b_{\phi_\Psi}(x_i^k)$$

Verteilung	θ	$\Psi(\theta)$	$P_{\theta,\Psi}(x)$	$d_\phi(x,y)$	S
Gauss	$\frac{\nu}{\sigma^2}$	$\frac{\sigma^2}{2}\theta^2$	$\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(\frac{(x-\nu)^2}{2\sigma^2}\right)$	$\frac{1}{2\sigma^2}(x-y)^2$	$\mathbb{R}$
Poisson	$\log\lambda$	$\exp(\theta)$	$\frac{\lambda^x\exp(-\lambda)}{\Gamma(x+1)}$	$x\log\left(\frac{x}{y}\right)-(x-y)$	$\mathbb{R}^+$
Exponential	$-\lambda$	$-\log(-\theta)$	$\lambda\exp(-\lambda x)$	$\frac{x}{y}-\log\left(\frac{x}{y}\right)-1$	$\mathbb{R}^+\setminus\{0\}$

Tab. 4.2: Verschiedene eindimensionale Verteilungen und die korrespondierenden Bregman-Divergenzen – siehe auch [BMDG05]

$$
\begin{aligned}
&= \arg\max_{j=1,\ldots,M} \log\left(\prod_{k=1}^{d}\exp\left(-d_{\phi_\Psi}(x_i^k,c_j^k)\right)b_{\phi_\Psi}(x_i^k)\right)\\
&= \arg\max_{j=1,\ldots,M} \sum_{k=1}^{d} -d_{\phi_\Psi}(x_i^k,c_j^k)+\sum_{k=1}^{d}\log\left(b_{\phi_\Psi}(x_i^k)\right)\\
&= \arg\min_{j=1,\ldots,M} \sum_{k=1}^{d} d_{\phi_\Psi}(x_i^k,c_j^k),
\end{aligned}
$$

wobei x_i^k, bzw. c_i^k die k-te Komponente des entsprechenden Vektors bezeichnet.

4.2.8 Zusammenfassung

Wir haben nun verschiedene Dimensionsbegriffe und Varianten für Schätzer dieser Dimensionen kennengelernt und außerdem den Concentration of Measure-Effekt erläutert.

- Wie wir gesehen haben, lässt sich die bekannte **Hausdorff-Dimension** (4.2) nicht gut numerisch schätzen.

- Man kann allerdings durch eine Vereinfachung des Hausdorff-Dimensionsbegriffes zur **Boxcounting-Dimension** (4.7) gelangen, für die man numerische Schätzer besser implementieren kann. Die Boxcounting-Dimension des Attraktors A ist auch diejenige, die wir schätzen wollen, um Takens' Delay-Einbettung anzugeben. Ein Nachteil der Boxcounting-Methode ist die Tatsache, dass beim numerischen Schätzen nicht auf das dem Attraktor zugrundeliegende Wahrscheinlichkeitsmaß eingegangen wird.

- Ein alternativer Schätzer, der dieses Problem umgeht, ist der **Korrelationsdimensionsschätzer** (4.9) nach Grassberger und Procaccia. Um hier das Problem von zeitlichen Korrelationen zu umgehen, wurde die Theilerkorrektur (4.22) und der Raum-Zeit-Seperationsplot (4.24) vorgestellt. Wie wir in Unterabschnitt 4.2.3 gesehen haben, ist es allerdings möglich, dass die Korrelationsdimension kleiner

sein kann als die Boxcounting-Dimension, weshalb wir die in Takens' Theorem benötigte Dimension $\lceil 2d_{cap} + 1 \rceil$ unterschätzen könnten. In hohen Dimensionen kann es sinnvoll sein die euklidische Norm im Argument der Heavyside-Funktion durch eine l_p-Norm für $p \in \mathbb{Q}^+,\ p < 2$ zu ersetzen.

- Eine alternative Herangehensweise an das Problem der Dimensionsfindung haben wir in der **PCA-Methode** 4.2.4 gefunden. Wir haben kurz erwähnt, dass ein Hauptproblem hierbei sein kann, dass sich – aufgrund eines zu großen Zeitparameters τ – der Prozess in alle Raumdimensionen ausdehnen kann.

- Unter der Voraussetzung, dass der Attraktor A eine Mannigfaltigkeit ist, eignet sich die **lokale PCA** (4.19) zum Schätzen der Mannigfaltigkeitendimension, sofern vorher ein geeignetes Clustering vorgenommen wurde. Je nach Art des Rauschens auf den Daten können verschiedene Bregman-Divergenzen ein sinnvolles Abstandsmaß darstellen.

- Zuletzt sei angemerkt, dass es durchaus noch viele andere Methoden gibt, die hier zum Einsatz kommen können, welche wir im Rahmen dieser Arbeit nicht näher betrachten wollen:
  - Einen weiteren PCA-ähnlichen Ansatz bietet z.B. die ***Kernel-PCA*** [SSM96], welche – ähnlich wie die lokale PCA – auch nichtlinear arbeitet.
  - Ein anderer Schätzer, der oftmals eingesetzt wird, ist der ***False-Nearest-Neighbours***-Schätzer, welcher die Einbettungsdimension solange erhöht, bis sich Abstände zwischen benachbarten Punkten nicht mehr signifikant ändern, siehe [KS04, Eng91].
  - Es sei noch darauf hingewiesen, dass oftmals auch ein schlichtes ***Trial-and-Error***-Prinzip durchgeführt wird, siehe [LV07]. Hier wird eine Einbettungsdimension n_1 gewählt und es wird versucht, den Prozess im $\mathbb{R}^{n_1}$ zu modellieren, bzw. vorherzusagen. Nun bettet man in eine größere Dimension $n_2 > n_1$ ein und beobachtet, ob sich die Vorhersagen signifikant verbessern. Liegt keine Verbesserung vor, geht man davon aus, dass die Dimension n_1 bereits genügt, um den Prozess einzubetten. Andernfalls, verfährt man analog mit n_2 und vergleicht die Vorhersagen im $\mathbb{R}^{n_2}$ mit denen in einem höherdimensionalen Raum, bis sich diese schlussendlich nicht mehr verbessern. Eine solche Methode, die auf Vorhersagen mittels ***Neuronalen Netzen*** beruht, findet sich in [KKKW95].

4.3 Wahl der Zeitschrittweite

In diesem Abschnitt wollen wir uns der Wahl des Zeitparameters (oder auch *Delay*) τ widmen. In der Theorie ist die konkrete Wahl von τ irrelevant. In Theorem 3.3 haben wir

gesehen, dass nahezu jede Wahl von τ zu einer korrekten Rekonstruktion des Attraktors führt.
In der Praxis liegt jedoch eine endliche Anzahl von Zahlen aus $\mathbb{R}$ als Zeitreihe vor und der zugrundeliegende Prozess ist nicht bekannt. Es gibt also keine Möglichkeit τ wirklich zu variieren. Jedoch können wir nur jeden T-ten Wert der Zeitreihe betrachten und erhalten somit eine neue Zeitreihe

$$(\nu_0, \nu_1, \nu_2, \ldots, \nu_N) \longrightarrow (\nu_0, \nu_T, \nu_{2T}, \ldots, \nu_{\hat{N}T}) \,, \tag{4.33}$$

wobei $\hat{N} = \lfloor \frac{N}{T} \rfloor$ ist. Die neue Zeitreihe ist allerdings nur noch von der Länge $\hat{N}$ und wir erhalten ein entsprechend gröberes Sampling des Attraktors.
Sprechen wir also bei einer konkret vorliegenden Zeitreihe von einer Variation von τ, so meinen wir nichts anderes als die Wahl von $T \in \mathbb{N}\setminus\{0\}$. Wir können somit die Zeitspanne der Beobachtungen bei einer konkreten Zeitreihe künstlich vergröbern. Sprechen wir im folgenden vom Zeitparameter τ bei einer vorliegenden Zeitreihe, so meinen wir immer $T \in \mathbb{N} \setminus \{0\}$.
Dass diese Vergröberung sinnvoll sein kann, merken wir, sobald unsere Zeitskala sehr fein aufgelöst ist. Wir haben dann eine Zeitreihe vorliegen, bei welcher aufeinanderfolgende Messungen stark korreliert sind und sich – bei glatten Prozessen wie wir sie in Takens' Theorem vorausgesetzt haben – nur wenig unterscheiden, siehe [KS04]. Somit erhalten wir für eine Einbettung in $\mathbb{R}^n$:

$$\nu_i \approx \nu_{i+1} \approx \cdots \approx \nu_{i+n-1} \approx \nu_{i+n} \Rightarrow \overbrace{(\nu_i, \ldots, \nu_{i+n-1})^T}^{x_i} \approx \overbrace{(\nu_{i+1}, \ldots, \nu_{i+n})^T}^{x_{i+1}} .$$

Hieraus ergeben sich folgende Probleme:

- Rekonstruierte Vektoren, die zeitlich benachbart sind, sind fast identisch. Wie wir schon beim Schätzer für die Korrelationsdimension gesehen haben, kann dies unerwünschte Effekte mit sich bringen.

- Da sich die Koordinaten der einzelnen Vektoren nur wenig unterscheiden, konzentrieren sich die Vektoren alle um die Raumdiagonale

  $$\left\{ \mathbf{z} = (z_1, \ldots, z_n)^T \in \mathbb{R}^n \mid z_1 = \cdots = z_n \right\} .$$

  Wir erhalten zwar somit noch eine Einbettung, jedoch ist der rekonstruierte Attraktor so gestaucht, dass sowohl Dimensionsschätzer als auch Vorhersagealgorithmen schlechte Ergebnisse produzieren. Wie man bereits ohne Probleme einsehen kann, führt dieses Phänomen dazu, dass die Dimension des Attraktors unterschätzt wird, da dieser durch die Stauchung die Struktur der 1-dimensionalen Raumdiagonale approximiert.

- Ein weiteres Problem, welches wir bereits in Abschnitt 4.1 angesprochen haben,

ist die Phase der Transienz. Die Trajektorie braucht eine gewisse Zeit, bis sie sich dem Attraktor annähert und approximativ den Bewegungsgleichungen folgt, die auf diesem gelten. Nehmen wir nun an, dass wir den Prozess in immer kleineren Zeitintervallen samplen, so erhalten wir beim Rekonstruieren immer mehr Punkte, die noch nicht nahe dem Attraktor liegen, sondern die transiente Phase der Trajektorie darstellen. Diese wollen wir allerdings in Dimensionsschätzern und Vorhersagen vernachlässigen.

Um diese Probleme zu umgehen, kann es also sinnvoll sein, den Zeitparameter τ zu vergrößern und es bei einer vorliegenden Zeitreihe nicht bei $\tau = 1$ zu belassen.
Es sei noch darauf hingewiesen, dass eine zu große Wahl von τ ebenfalls zu Problemen führen kann:

- Wir haben in Gleichung (4.33) bereits gesehen, dass sich durch eine Wahl $\tau > 1$ auch die Anzahl der vorliegenden rekonstruierten Vektoren verkleinert.
- Für eine zu große Wahl von τ ergibt sich das gegenteilige Phänomen zur Konzentration der Vektoren um die Raumdiagonale: Die rekonstruierten Punkte scheinen völlig unkorreliert zu sein und man erhält zwar eine Rekonstruktion des Attraktors, jedoch sieht diese sehr kompliziert und vor allem unstrukturiert aus. Die Vorhersagealgorithmen können auf solchen Mengen keine guten Modelle für die wirkliche Dynamik des rekonstruierten Prozesses liefern.

Zwei Verfahren, um – anhand einer konkreten Zeitreihe – eine heuristische Schätzung für eine gute Wahl von τ zu erhalten, wollen wir in diesem Abschnitt näher betrachten. Leider gibt es in der Theorie keinerlei Aussagen über einen "perfekten" *Time-Lag* τ. [KS04] weist darauf hin, dass jegliche Verfahren zum Schätzen dieser Größe lediglich Heuristiken sind.

4.3.1 Die Autokorrelationsfunktion

Eine relativ simple Methode zum heuristischen Schätzen eines adäquaten Delays τ liefert die Autokorrelationsfunktion. Diese misst die lineare Beziehung der Zeitreihe zu einer zeitversetzten Variante.
Liegt eine Observable o vor, die auf einen Prozess angewendet wird, so nennen wir die resultierende Zeitreihe ν. Also gilt

$$\nu_t(\mathbf{z_0}) = o(\phi_t(\mathbf{z_0})),\ t \in \mathbb{R}^+$$

für zeitkontinuierliche Prozesse und

$$\nu_t(\mathbf{z_0}) = o(\phi^t(\mathbf{z_0})),\ t \in \mathbb{N}$$

für zeitdiskrete Prozesse. Wir betrachten nun die Zeitreihe als Zufallsvariable abhängig

von der Zeit t. Diese Sichtweise ist gerechtfertigt, wenn die Anfangsbedingung unbekannt ist, siehe [KS04].

Definition 4.11 [Autokorrelationsfunktion]
Ist die Zeitreihe $\nu_t : M \to \mathbb{R}$ zum Zeitpunkt t nach dem Maß μ_t verteilt und gilt weiterhin Stationärität ($\mu_t \equiv \mu \ \forall\ t \in \mathbb{R}^+$(bzw. $\forall\ t \in \mathbb{N}$)), so nennen wir

$$C(\tau) := \frac{1}{\sigma_\mu^2} \cdot \mathbb{E}_\mu \left[(\nu_t - \mathbb{E}_\mu[\nu_t])(\nu_{t-\tau} - \mathbb{E}_\mu[\nu_t])\right] = \frac{\mathbb{E}_\mu[\nu_t \nu_{t-\tau}] - \mathbb{E}_\mu[\nu_t]^2}{\sigma_\mu^2} \tag{4.34}$$

*die **Autokorrelationsfunktion** der Zeitreihe.*

Die Autokorrelationsfunktion misst die Kovarianz zwischen der Zeitreihe und einer zeitversetzten Variante. Dieser Wert wird mittels der Varianz der Zeitreihe normiert.
Die Forderung der Stationarität ist bei uns grundsätzlich erfüllt, da dies gleichbedeutend damit ist, dass sich die Bewegungsgleichungen (also der Diffeomorphismus ϕ in Definition 2.1, bzw. das Vektorfeld $\mathbf{X}$ in Definition 2.2) des Prozesses nicht mit der Zeit ändern können. Eine solche Änderung haben wir bei der Definition des Begriffs der Zeitreihe nicht zugelassen.
Wir wollen nun das Verhalten der Autokorrelationsfunktion näher betrachten und diverse Eigenschaften erläutern:

- Wie wir sofort sehen, gilt **für unkorrelierte Zufallsvariablen** ν_t und $\nu_{t-\tau}$

$$C(\tau) = \frac{\mathbb{E}_\mu[\nu_t] \cdot \mathbb{E}_\mu[\nu_{t-\tau}] - \mathbb{E}_\mu[\nu_t]^2}{\sigma_\mu^2} = \frac{\mathbb{E}_\mu[\nu_t] \cdot \mathbb{E}_\mu[\nu_t] - \mathbb{E}_\mu[\nu_t]^2}{\sigma_\mu^2} = 0 \ .$$

- **Eine vollständige Kopplung** $\nu_t = \pm\nu_{t-\tau}$ widerrum, führt direkt zu

$$C(\tau) = \frac{\mathbb{E}_\mu[\nu_t \cdot (\pm\nu_t)] - \mathbb{E}_\mu[\nu_t]^2}{\sigma_\mu^2} = \frac{\pm\mathbb{E}_\mu[\nu_t \cdot \nu_t] - \mathbb{E}_\mu[\nu_t]^2}{\sigma_\mu^2} \overset{(*)}{=} \pm 1 \ .$$

  Für $\nu_t = \nu_{t-\tau}$ ist $(*)$ aufgrund der Definition der Varianz $\sigma_\mu^2 := \mathbb{E}_\mu[\nu_t^2] - \mathbb{E}_\mu[\nu_t]^2$ trivial, doch die Gleichung gilt auch für den Fall negativer Korrelation, da dann

$$\mathbb{E}_\mu[\nu_t] = \mathbb{E}_\mu[\nu_{t-\tau}] = \mathbb{E}_\mu[-\nu_t] = -\mathbb{E}_\mu[\nu_t]$$

  gilt und somit $\mathbb{E}_\mu[\nu_t] = 0$ ist.

  Eine beliebige Kopplung der Art $\nu_t = f(\nu_{t-\tau})$ für $f : \mathbb{R} \to \mathbb{R}$ ist im Allgemeinen wegen $\mu_t = \mu_{t-\tau}$ nicht möglich, da somit jegliche Momente $\mathbb{E}[\nu_t^k]$ und $\mathbb{E}[\nu_{t-\tau}^k]$ gleich sein müssen.

- Dass die Autokorrelation lediglich **lineare Zusammenhänge** misst, wird klar, wenn man den rechten Term in Gleichung (4.34) betrachtet: Hier sehen wir, dass

wir lediglich das erste Moment der Produktzufallsvariablen $\nu_t \nu_{t-\tau}$ betrachten. Für Beziehungen höherer Ordnung müssten wir auch höhere Momente betrachten.

- Wollen wir im Fall einer vorliegenden **endlichen Zeitreihe** die Autokorrelation schätzen, müssen wir wie folgt vorgehen:

  Da wir in diesem Fall keinerlei Kenntnis über den Prozess oder über das Maß μ haben, können wir die Autokorrelation nur anhand der Zeitreihe für $\tau \in \mathbb{N}$ schätzen:

$$C(\tau) \approx \hat{C}(\tau) := \frac{1}{\hat{\sigma}^2} \cdot \frac{1}{N-\tau-1} \sum_{i=\tau+1}^{N} \left(\left(\nu_i - \frac{1}{N} \sum_{j=1}^{N} \nu_j \right) \left(\nu_{i-\tau} - \frac{1}{N} \sum_{j=1}^{N} \nu_j \right) \right). \tag{4.35}$$

  Hierbei ist $\hat{\sigma}^2$ die empirische Varianz der Zeitreihe:

$$\hat{\sigma}^2 = \frac{1}{N-1} \sum_{i=1}^{N} \left(\nu_i - \frac{1}{N} \sum_{j=1}^{N} \nu_j \right)^2.$$

  Es sei angemerkt, dass der Schätzer $\hat{C}(\tau)$ nicht erwartungstreu ist. Für eine Betrachtung des Konvergenzverhaltens von $\hat{C}(\tau)$ gegen $C(\tau)$ sei auf [HZZGH82] verwiesen. Bei genauerer Inspektion der Ergebnisse dort, ist festzustellen, dass im Allgemeinen mit fast sicherer Konvergenz mit einer Rate von $O(\sqrt{N-\tau})$ zu rechnen ist. Dies soll hier allerdings nicht näher ausgeführt werden.

Nun ist unser Ziel, mithilfe der Autokorrelationsfunktion einen guten Wert für $\tau \in \mathbb{N} \setminus \{0\}$ zu finden. In der Praxis wählt man meistens $\tau \in \mathbb{N}$ als die kleinste natürliche Zahl, sodass

$$\hat{C}(\tau) \leq \frac{1}{e} \tag{4.36}$$

gilt, wobei e hier die Euler'sche Zahl bezeichnet. Diese Strategie erwies sich oftmals als eine gute Herangehensweise, siehe [KS04].
Der Wert, welcher von der Autokorrelationsfunktion unterschritten werden soll, ist allerdings willkürlich. [Eng91] schlägt z.B. vor $\frac{1}{e}$ durch $\frac{1}{2}$ zu ersetzen.
Die Heuristik, nach der wir schätzen, besagt lediglich, dass wir keine stark korrelierten Daten wollen, aber genausowenig völlig unkorrelierte Punkte. Gehen wir von einem glatten Prozess aus, schränkt Letzteres die Wahl einer guten heuristischen Schranke auf das positive Intervall $(0, 1)$ ein.
Es sei noch darauf hingewiesen, dass ein Schätzen des Zeitparameters τ nach der Autokorrelationsfunktion eng verwandt mit der in [BK85] für die PCA vorgeschlagenen Methode ist, bei welcher das Fourierspektrum der Zeitreihe betrachtet wird. Dieser Zusammenhang drückt sich im *Wiener-Chintchin-Theorem* [Chi34] aus, welches besagt, dass

$$F(C)(\omega) = |F(\nu)(\omega)|^2$$

gilt, wobei F die Fouriertransformierte einer Funktion bezeichnet. Wir gehen hier davon aus, dass der Anfangszustand $\mathbf{z_0}$ fest, aber unbekannt ist und wollen ν nur als Funktion von t auffassen.

4.3.2 Mutual Information

Da das Schätzen von τ anhand der Autokorrelationsfunktion lediglich lineare Zusammenhänge in den Komponenten der Zeitreihe beachtet, schlugen Fraser und Swinney in [FS86] eine andere Herangehensweise vor.
Wir führen nun den Begriff der *Shannon-Entropie* [SW98] als Maß für die Unsicherheit des Ausgangs eines Zufallsexperiments ein, um die Mutual Information-Methode zu motivieren. Im Folgenden soll $0 \log 0 := 0$ gelten.

Definition 4.12 [(Shannon-)Entropie]
Sei $(S = \{s_i\}, \mathcal{P}(S), \mu = (\mu_i))$ ein diskreter Wahrscheinlichkeitsraum und X eine Zufallsvariable mit Bild in S, die nach μ verteilt ist. Dann bezeichnet

$$H(X) := -\sum_{s_i \in S} \mu_i \log \mu_i = \mathbb{E}_\mu [\log \mu] \tag{4.37}$$

*die **diskrete Entropie** von X, bzw. μ.*

Eine große Entropie ist gleichbedeutend mit großer Ungewissheit über den Ausgang eines Zufallsexperiments mit X.
Liegen nun zwei Zufallsvariablen X, Y mit demselben Wertebereich S vor, deren gemeinsame Verteilung dem Maß $\gamma := (\gamma_{ij})_{i,j=1}^K$ folgt, so lässt sich die Entropie der gemeinsamen Verteilung durch

$$H(X,Y) = -\sum_{s_i, s_j \in S} \gamma_{ij} \log \gamma_{ij}$$

angeben und mit $\gamma_{i*} := \sum_{s_j \in S} \gamma_{ij}$ gilt

$$H(X) = -\sum_{s_i \in S} \gamma_{i*} \log \gamma_{i*}.$$

Analog lässt sich $H(Y)$ ausrechnen. Sind wir im Fall der Kenntnis von X an der Entropie für Y interessiert, so ergibt sich diese als Erwartungswert über die Entropie der bedingten Wahrscheinlichkeit von Y gegeben $X = s_i$:

$$H(Y|X) = -\sum_{s_i \in S} \gamma_{i*} \sum_{s_j \in S} \frac{\gamma_{ij}}{\gamma_{i*}} \log \frac{\gamma_{ij}}{\gamma_{i*}} = -\sum_{s_i, s_j \in S} \gamma_{ij} \log \frac{\gamma_{ij}}{\gamma_{i*}}.$$

Wie man unmittelbar sieht, gilt $H(X,Y) = H(X) + H(Y|X) = H(Y) + H(X|Y)$, was

auch zu erwarten war. Die Mutual Information ist nun durch

$$I(X,Y) := H(X) - H(X|Y) = H(Y) - H(Y|X) = \sum_{s_i,s_j \in S} \gamma_{ij} \log \frac{\gamma_{ij}}{\gamma_{i*}\gamma_{*j}}$$

definiert. Dies ist interpretierbar als die Unsicherheit über den Ausgang eines Zufallsexperiments mit X, wobei aber jegliche Informationen, die der Ausgang eines Experiments mit Y über X liefert, bekannt sind.
Um die Mutual Information im Zeitreihenkontext nutzen zu können, muss die Zeitreihe zunächst vorverarbeitet werden:
Wir unterteilen den Wertebereich der Zeitreihe in K disjunkte, gleichlange Intervalle I_k

$$\bigcup_{k=0}^{K-1} \overbrace{\left[\min_t \nu_t + \frac{k}{K}(\max_t \nu_t - \min_t \nu_t),\ \min_t \nu_t + \frac{k+1}{K}(\max_t \nu_t - \min_t \nu_t)\right)}^{I_k}.$$

Damit wir eine Überdeckung erhalten, setzen wir noch $I_{K-1} := \overline{I_{K-1}}$.
Ist ν_t dann nach dem Maß μ verteilt, erhalten wir

$$\mu(I_k) = \mathbb{P}[\nu_t \in I_k].$$

Weiterhin wollen wir mit γ_τ die gemeinsame Verteilung der Zeitreihe und der versetzten Zeitreihe bezeichnen. Es gilt also

$$\gamma_\tau(I_i, I_j) := \mathbb{P}[\nu_t \in I_i \text{ und } \nu_{t-\tau} \in I_j].$$

Definition 4.13 [Time-Delayed Mutual Information]
Sei ϵ die Länge $\lambda_{\mathbb{R}}(I_0)$ eines der Intervalle I_k. Wir nennen

$$M_\epsilon(\tau) := \sum_{i,j=0}^{K-1} \gamma_\tau(I_i, I_j) \log \frac{\gamma_\tau(I_i, I_j)}{\mu(I_i)\mu(I_j)} \tag{4.38}$$

$$= \sum_{i,j=0}^{K-1} \gamma_\tau(I_i, I_j) \log \gamma_\tau(I_i, I_j) - 2\sum_{k=0}^{K-1} \mu(I_k) \log \mu(I_k) \tag{4.39}$$

Time-Delayed Mutual Information *zum Parameter τ.*

Wir wollen das Verhalten und die Eigenschaften der Mutual Information genauer untersuchen:

- Wie wir sehen, stellt diese ein Maß für die Korreliertheit des Prozesses zu einer zeitversetzten Variante dar. **Für unkorrelierte Variablen** ν_t und $\nu_{t-\tau}$ erhalten

wir $\gamma_\tau = \mu \otimes \mu$ und somit

$$\begin{aligned} M_\epsilon(\tau) &= \sum_{i,j=0}^{K-1} \mu(I_i)\mu(I_j) \log\left(\mu(I_i)\mu(I_j)\right) - 2\sum_{k=0}^{K-1} \mu(I_k) \log \mu(I_k) \\ &= \overbrace{\sum_{j=0}^{K-1} \mu(I_j)}^{=1} \cdot \sum_{i=0}^{K-1} \mu(I_i) \log \mu(I_i) + \overbrace{\sum_{i=0}^{K-1} \mu(I_i)}^{=1} \cdot \sum_{j=0}^{K-1} \mu(I_j) \log \mu(I_j) \\ &\quad -2\sum_{k=0}^{K-1} \mu(I_k) \log \mu(I_k) \\ &= \sum_{i=0}^{K-1} \mu(I_i) \log \mu(I_i) + \sum_{j=0}^{K-1} \mu(I_j) \log \mu(I_j) - 2\sum_{k=0}^{K-1} \mu(I_k) \log \mu(I_k) \\ &= 0\ . \end{aligned}$$

- **Für vollständig korrelierte** Zufallsvariablen $\nu_t = \pm\nu_{t-\tau}$ gilt für $A, B \subset \mathbb{R}$

$$\gamma_\tau(A, B) = \mu(A \cap \pm B)\ .$$

Die Mutual Information ist dann

$$\begin{aligned} M_\epsilon(\tau) &= \sum_{i,j=0}^{K-1} \mu(I_i \cap \pm I_j) \log \mu(I_i \cap \pm I_j) - 2\sum_{k=0}^{K-1} \mu(I_k) \log \mu(I_k) \\ &\overset{(*)}{=} \sum_{i=0}^{K-1} \mu(I_i) \log \mu(I_i) - 2\sum_{k=0}^{K-1} \mu(I_k) \log \mu(I_k) \\ &= -\sum_{k=0}^{K-1} \mu(I_k) \log \mu(I_k), \end{aligned}$$

und somit genau gleich der Shannon-Entropie bezüglich μ. Für positive Korrelation ist $(*)$ sofort klar, doch dies gilt auch für negative Korrelation, da hierbei $\max \nu_t = -\min \nu_t$ sein muss und unsere Intervallschachtelung symmetrisch um 0 ist. Dies führt dazu, dass es zu jedem I_i genau ein I_j gibt, sodass $I_i = -I_j$ ist.

- Da bei der Berechnung eine Aufteilung in Intervalle und die Kalkulation der Shannon-Entropie zur Prüfgröße führen, ist es hier möglich auch **nichtlineare Zusammenhänge** in den Daten zu finden.

- Wollen wir die **Mutual Information in der Praxis** ausrechnen, erstellen wir einfach ein- und zweidimensionale Histogramme, welche uns die empirischen Wahr-

scheinlichkeiten

$$\begin{array}{rcl} \hat{\mu}(I_k) & = & \frac{1}{N}\#\{n \in \{1,\ldots,N\} \mid \nu_n \in I_k\} \\ \hat{\gamma}_\tau(I_i, I_j) & = & \frac{1}{N-\tau}\#\{n \in \{\tau+1,\ldots,N\} \mid \nu_n \in I_i \text{ und } \nu_{n-\tau} \in I_j\} \end{array} \tag{4.40}$$

anhand der Zeitreihe schätzen lassen. Die empirischen Verteilungen konvergieren nach dem Gesetz der großen Zahlen mit der Rate $O(\sqrt{N-\tau})$ gegen die zugrundeliegende Wahrscheinlichkeitsverteilung. Zudem konvergiert auch die Mutual Information der empirischen Verteilungen (auf einem endlichen Zustandsraum) mit der Rate $O(\sqrt{N-\tau})$ gegen die Mutual Information der zugrundeliegenden Wahrscheinlichkeitsverteilungen. Hierzu sei auf Abschnitt 4.1 in [AK01] verwiesen.

Es wird nun empfohlen $\tau \in \mathbb{N}$ so zu wählen, dass die mittels (4.40) geschätzte Mutual Information dort das erste Minimum annimmt, siehe [KS04, FS86]. Im Sinne der Shannon-Entropie beinhaltet $\nu_{t-\tau}$ dann die meiste Information über ν_t.
[KS04] zeigt, dass der Grenzwert der Mutual Information für ϵ gegen 0 nicht existieren muss. Es wird dort empfohlen, ein $\epsilon > 0$ festzusetzen und mit diesem die Mutual Information auszurechnen. Weiterhin wird angeführt, dass eine große Wahl von ϵ keinen zwangsläufigen Güteverlust unseres Schätzers zu bedeuten hat, da wir nicht am konkreten Wert, sondern nur am qualitativen Verlauf der Mutual Information interessiert sind. Es heißt dort lediglich, dass eine klare τ-Abhängigkeit aus der Mutual Information ablesbar sein sollte.
Es sei noch angemerkt, dass es auch möglich ist, an die Daten angepasste, nichtuniforme Intervalllängen zu wählen. Eine Methode hierzu findet sich im Originalartikel [FS86]. Wir orientieren uns bei unseren Experimenten in Kapitel 7, am einfacheren, in [KS04] vorgeschlagenen Algorithmus.

4.3.3 Zusammenfassung

Wir haben zwei heuristische Verfahren zum Schätzen des Zeitparameters τ kennengelernt. Für beide Verfahren wurde über die Beispiele eines unkorrelierten und eines determinierten Prozesses motiviert, wieso sie eine sinnvolle Größe für die Korrelation des Prozesses mit einer zeitversetzten Kopie liefern.
Es wurden Schätzer vorgestellt, welche es ermöglichen, die beiden Größen anhand einer endlichen Zeitreihe zu approximieren.

- Das erste Verfahren macht sich die **Autokorrelationsfunktion** zunutze, mit welcher es möglich ist, lineare Zusammenhänge zwischen zeitversetzten Daten innerhalb der Zeitreihe zu finden. Bei dieser Methode wird der erste Zeitpunkt gesucht, an welchem der Wert der Autokorrelationsfunktion eine gewisse Schranke unterschreitet. Es wurde angemerkt, dass diese Methode mit der in [BK85] über das Wiener-Chintchin-Theorem verwandt ist.

- Das zweite vorgestellte Verfahren berechnet die **Mutual Information** der Zeitrei-

he und benutzt diese als Maß für die Korreliertheit des Prozesses mit einer zeitversetzten Kopie. Um hier eine gute Schätzung für τ zu finden, wird das erste Minimum der Mutual Information in Abhängigkeit der Zeit gesucht.

- Es sei noch angemerkt, dass – ähnlich wie bei den Dimensionsschätzern (vgl. Abschnitt 4.2.8) – oftmals auch **Trial-and-Error**-Verfahren benutzt werden, um den Zeitparameter τ zu bestimmen, siehe [KS04]: Der Prozess wird zunächst mit dem Zeitparameter τ_1 eingebettet. Liefert eine Vorhersage anhand dieser Einbettung zufriedenstellende Ergebnisse, so geht man davon aus, dass τ_1 eine gute Wahl war, andernfalls wählt man $\tau_2 > \tau_1$ und verfährt analog, bis man ein passendes τ_k gefunden hat.

5 Vorhersage der Zeitreihe mit dünnen Gittern

Es wurde eine Methode vorgestellt, die Trajektorie eines Prozesses anhand endlich vieler Zeitreihen-Werte zu rekonstruieren. Somit haben wir die Möglichkeit, Aussagen über die Struktur und Dimension des Attraktors des Prozesses zu treffen.
In diesem Kapitel wird eine Vorgehensweise vorgestellt, um den zukünftigen Verlauf der Zeitreihe und des zugrundeliegenden Prozesses vorherzusagen.
Wir treffen nun keine Unterscheidung mehr zwischen diskreten und kontinuierlichen Prozessen und gehen davon aus, dass eine endliche, zeitlich-indizierte Folge von reellen Werten vorliegt.
Ist $(s_i)_{i=1}^N$ die Zeitreihe, welche in $\mathbb{R}^d$ eingebettet wird, so bezeichnen wir im Folgenden mit

$$\mathbf{x_i} = \begin{pmatrix} s_i \\ s_{i-1} \\ \vdots \\ s_{i-d+2} \\ s_{i-d+1} \end{pmatrix} \in \mathbb{R}^d$$

den Zustand des rekonstruierten Prozesses zum Zeitpunkt $i \in \mathbb{N}$.
Es sei darauf hingewiesen, dass eine Vorhersage der Zeitreihe äquivalent zu einer Vorhersage des rekonstruierten Prozesses ist. Dies liegt am Prinzip der Delay-Einbettung.[1] Ziel ist es, aufgrund der Kenntnis des aktuellen Zustands des rekonstruierten, d-dimensionalen Prozesses den Folgezustand vorherzusagen. Da dieser durch den aktuellen Zustand determiniert ist, gibt es einen funktionalen Zusammenhang:

$$\exists\ \mathbf{G} : \mathbb{R}^d \to \mathbb{R}^d\ :\ \mathbf{x_{i+1}} = \mathbf{G}(\mathbf{x_i})\ \forall\ i \in \mathbb{N}. \tag{5.1}$$

Mit $\mathbf{G}$ kann nicht nur die bisher vorliegende Zeitreihe $\mathbf{x_i}$ für $i \leq N$, sondern auch die Weiterentwicklung vorhergesagt werden.
Da durch die Kenntnis von $\mathbf{x_i}$ bereits $d-1$ der d Koordinaten von $\mathbf{x_{i+1}}$ bekannt sind,

[1] Dies ist nur dann der Fall, wenn die Vorhersage des unmittelbar folgenden Zeitreihenwertes gemeint ist.

ist eine Umformulierung möglich:

$$\exists\, g : \mathbb{R}^d \to \mathbb{R} \;:\; \mathbf{x_{i+1}} = \begin{pmatrix} g(\mathbf{x_i}) \\ s_i \\ \vdots \\ s_{i-d+3} \\ s_{i-d+2} \end{pmatrix} \quad \forall i \in \mathbb{N} \tag{5.2}$$

Um die Zeitreihe und den Prozess vorhersagen zu können, muss ein solches g gefunden, bzw. gut approximiert werden.
g beschreibt ein skalares Feld über $\mathbb{R}^d$ und somit eine d-dimensionale Hyperfläche im $\mathbb{R}^{d+1}$. Eine geometrische Interpretation unseres Ziels g zu rekonstruieren, ist, eine Fläche im $\mathbb{R}^{d+1}$ zu finden, welche die Punktwolke $\begin{pmatrix} s_{i+1} \\ \mathbf{x_i} \end{pmatrix}$ approximiert.

5.1 Regularisiertes Fehlerfunktional

Um g anhand der vorliegenden endlichen Zeitreihe durch eine Funktion f zu approximieren, minimieren wir ein Fehlerfunktional über einer Menge von Funktionen. Eine ausführliche Auseinadersetzung mit diesem Thema ist in [SS02] zu finden. In dieser Arbeit wird das Fehlerfunktional aus [Gar04] verwendet, welches im Folgenden kurz motiviert wird.

5.1.1 Datenfehler

Um schlechte Vorhersagen im Fehlerfunktional zu bestrafen, wird zunächst eine Kostenfunktion

$$c : \mathbb{R} \times \mathbb{R} \to \mathbb{R}^+ \tag{5.3}$$

definiert, wobei

$$c(s_{i+1}, f(\mathbf{x_i}))$$

ein Maß für den Fehler sein soll, der entsteht, wenn $f(\mathbf{x_i})$ als Approximation für s_{i+1} verwendet wird.
Weiterhin sei $p : \mathbb{R}^d \times \mathbb{R} \to [0,1]$ die Verteilung, welche die Zeitreihenwerte bestimmt. Im störungsfreien Fall (5.2), gilt also

$$p(\mathbf{x_i}, y) = \delta_{g(\mathbf{x_i})}(y).$$

Ziel ist es dann das folgende Funktional – und somit die erwarteten Kosten – zu minimieren:

$$\mathcal{R}(f) := \mathbb{E}[c(y, f(\mathbf{x}))] = \int_{\mathbb{R}\times\mathbb{R}} c(y, f(\mathbf{x}))\; dp(\mathbf{x}, y).$$

Da die Verteilung der Daten nicht bekannt ist, wird p durch die empirische Dichte approximiert:

$$p \approx \frac{1}{N-d} \sum_{i=d}^{N-1} \delta_{\mathbf{x_i}} \otimes \delta_{s_{i+1}}$$

Daraus ergibt sich das Minimierungsproblem

$$\frac{1}{N-d} \sum_{i=d}^{N-1} \int_{\mathbb{R}\times\mathbb{R}} c(y, f(\mathbf{x}))\; d(\delta_{\mathbf{x_i}} \otimes \delta_{s_{i+1}})(\mathbf{x}, y) = \underbrace{\frac{1}{N-d} \sum_{i=d}^{N-1} c(s_{i+1}, f(\mathbf{x_i}))}_{=:R_{\text{data}}(f)} \quad \xrightarrow{f\in\Gamma} \quad \min\,! \tag{5.4}$$

Der Raum Γ, über welchem minimiert wird, ist noch zu spezifizieren.
In Abschnitt 3.3 von [SS02] wird gezeigt, dass ein f, welches R_{data} minimiert, den Likelihood der vorliegenden Zeitreihe maximiert, sofern man annimmt, dass die Daten lediglich additiv verrauscht sind und

$$c(y, f(\mathbf{x})) = -\log p_\xi(y - f(\mathbf{x})) \tag{5.5}$$

gilt. p_ξ ist hierbei die Dichte des Rauschens. Dies wird analog zur im Unterabschnitt 4.2.7 beschriebenen Likelihood-Maximierung der Clustermittelpunkte gezeigt, siehe auch Gleichung (4.30).
Geht man nun beispielsweise davon aus, dass die Daten keine Störung tragen, so gilt $p_\xi = \delta_0$ und somit

$$c(y, f(\mathbf{x})) = \begin{cases} 0 & \text{falls } f(\mathbf{x}) = y \\ \infty & \text{sonst} \end{cases},$$

sofern man (5.5) voraussetzt.
In der Praxis wird oftmals von normalverteiltem Rauschen ausgegangen, was zu

$$p_\xi(z) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{z^2}{2}\right)$$

führt. Bei einer Likelihood-Maximierung erhält man somit:

$$c(y, f(\mathbf{x})) = -\left(\log\left(\frac{1}{\sqrt{2\pi}}\right) - \frac{(y - f(\mathbf{x}))^2}{2}\right). \tag{5.6}$$

Vernachlässigt man die Konstanten, so erhält man durch Einsetzen von (5.6) in (5.4):

$$R_{\text{data}}(f) = \frac{1}{N-d} \sum_{i=d}^{N-1} (s_{i+1} - f(\mathbf{x_i}))^2 \quad \xrightarrow{f\in\Gamma} \quad \min\,!. \tag{5.7}$$

Dieser Term stellt ein Maß für die Abweichung der Funktion f von den vorliegenden

Zeitreihendaten dar.
Es sei angemerkt, dass andere Kostenfunktionen zu Problemen führen können, die mit nichtlinearen Minimierungsverfahren behandelt werden müssen. Hierzu sei auf [SS02] verwiesen.

5.1.2 Regularisierung

Wird lediglich $R_{\text{data}}(f)$ minimiert, führt dies für die meisten Funktionenräume Γ zu einem schlecht gestellten Problem, da es mehrere Lösungen gibt.
Um dies zu vermeiden, schlug Tikhonov [Tik63] das Hinzufügen eines Regularisierungsterms vor. Dieser stellt gewisse Glattheitsanforderungen an die Funktion f und schränkt somit auch den Funktionenraum Γ ein.
Eine ausführliche Abhandlung dieses Themas ist u.a. in [Tik63, Gar04, SS02] zu finden.
Betrachtet wird das Minimierungsproblem

$$R(f) := R_{\text{data}}(f) + \lambda\Phi(f) \quad \xrightarrow{f\in\Gamma} \quad \min\,!, \tag{5.8}$$

wobei $\Phi : \Gamma \to \mathbb{R}$ ein Glattheitsfunktional darstellt.
Der Parameter $\lambda \in [0, \infty)$ gibt an, wie der Datenfehler und das Glattheitsfunktional balanciert werden.
In [Gar04] wurde $\Phi = ||\nabla f||_{L_2}^2 = |f|_{H^1}^2$ gesetzt, womit $|f|_{H^1_{mix}}^2$ approximiert werden soll. Im Rahmen dieser Arbeit wollen wir zusätzlich $\Phi(f) = |f|_{H^1_{mix}}^2$ verwenden. Im Folgenden soll kurz motiviert werden, weshalb eine Betrachtung der Sobolevräume mit gemischten dominierenden Ableitungen sinnvoll ist.
Zum Nachschlagen des Konzepts der Sobolevräume seien [Bra03, Gri06b] empfohlen.

Hilberträume mit reproduzierendem Kern

Die folgenden Ausführungen orientieren sich an [SS02] und sollen lediglich einen kurzen Einblick in das Themengebiet liefern.

Definition 5.1 [Hilberträume mit reproduzierendem Kern (RKHS)]
*Sei $(\mathcal{H}, \langle\cdot,\cdot\rangle)$ ein Hilbertraum von Funktionen $f : X \to \mathbb{R}$. Dann nennt man $(\mathcal{H}, \langle\cdot,\cdot\rangle)$ einen **RKHS**, falls eine symmetrische, positiv semi-definite Funktion $k : X \times X \to \mathbb{R}$ existiert, sodass gilt:*

1. *k ist reproduzierend:*
$$\langle u, k(x,\cdot)\rangle = u(x)\ \forall u \in \mathcal{H}.$$
2. *k spannt $\mathcal{H}$ auf:*
$$\mathcal{H} = \operatorname{span}\overline{\{k(x,\cdot) \mid x \in X\}}.$$

*Eine solche Funktion k nennt man **reproduzierenden Kern** von $\mathcal{H}$.*

In [SS02] werden viele wesentliche und interessante Aspekte von RKHS untersucht. In dieser Arbeit wird auf das Meiste verzichtet und lediglich die Wahl eines anderen Regularisierungsterms motiviert.
Eine wichtige Eigenschaft der RKHS ist, dass alle Punktauswertungen – also die Funktionale $u \to u(x)$ mit $x \in X$ – stetig sind.
Eine alternative Wahl für das Glattheitsfunktional aus (5.8) ist

$$\Phi(f) = ||f||^2_{\mathcal{H}},$$

wobei $\mathcal{H}$ ein Hilbertraum mit reproduzierendem Kern k ist.
Eine solche Regularisierung führt dazu, dass sich eine Lösung $\hat{f}$ von (5.8) mittels des *Repräsenter-Theorems* wie folgt darstellen lässt:

$$\hat{f}(\mathbf{x}) = \sum_{i=d}^{N-1} \alpha_i k(\mathbf{x_i}, \mathbf{x}),\ \alpha_i \in \mathbb{R}. \tag{5.9}$$

Obwohl ein Minimierungsproblem in einem unendlich-dimensionalen Hilbertraum gelöst wird, ist die Lösung lediglich eine Linearkombination endlich vieler, in den Datenpunkten verankerter Kernfunktionen. Dies macht man sich u.a. bei *Support-Vektor-Maschinen* zu Nutze, siehe [SS02].
Es sei angemerkt, dass der Hilbertraum $\mathcal{H}$ somit direkt den Unterraum span $\{k(\mathbf{x_i}, \mathbf{x})\}$ bestimmt, aus welchem die Lösungen für (5.8) stammen.
Weiterhin kann man bei einer solchen Regularisierung eine bestimmte Approximationsgüte sicherstellen [Lou08]. Weitere Grundlagen über die Theorie der Regularisierung mit reproduzierenden Kernen finden sich u.a. in [SS02, Bra09, Gar04].
Wir wollen nun mit S einen beliebigen Differentialoperator bezeichnen. Damit sich die Eigenschaften der Regularisierung mittels RKHS auf den Fall einer Regularisierung der Form

$$\Phi(f) = ||S(f)||^2_{L_2}$$

übertragen lassen, müssen wir diese in den Kontext der reproduzierenden Kerne übertragen.

H^s_{mix}-Sobolevräume als Hilberträume mit reproduzierendem Kern

Der in [Gar04] verwendete Operator $S = \nabla$ führt zur Seminorm des Sobolevraums H^1:

$$\Phi(f) = ||\nabla f||^2_{L_2} = \sum_{i=1}^{d} ||\frac{\partial}{\partial x_i} f||^2_{L_2} = |f|^2_{H^1}.$$

Die Sobolevräume H^m sind im Allgemeinen keine RKHS, da die Punktauswertungen – nach dem Lemma von Sobolev [FSZ01] – für $2m \leq d$ nicht stetig sein müssen.

Ein anderer Differentialoperator führt zu

$$\Phi(f) = \sum_{|\mathbf{a}|_\infty \leq m} \left|\left| \frac{\partial^{a_1}}{\partial x_1^{a_1}} \cdots \frac{\partial^{a_d}}{\partial x_d^{a_d}} f \right|\right|_{L_2}^2 = ||f||_{H^m_{mix}}^2. \tag{5.10}$$

$m \in \mathbb{N} \setminus \{0\}$ ist hierbei beliebig und $\mathbf{a} = (a_1, \ldots, a_d) \in \mathbb{N}^d$ stellt einen Multiindex dar. $\mathbf{a}$ wird in diesem Fall in der l_∞-Norm gemessen:

$$|\mathbf{a}|_\infty := \max_{i=1,\ldots,d} |a_i|. \tag{5.11}$$

Dies führt in der obigen Gleichung zum Sobolevraum mit dominierender gemischter Ableitung H^m_{mix}. Würde man stattdessen die l_1-Norm

$$|\mathbf{a}|_1 := \sum_{i=1}^{d} |a_i| \tag{5.12}$$

zugrunde legen, so würde man den Standard-Sobolevraum H^m erhalten.
Der Raum H^m_{mix} ist ein RKHS (siehe [SS02, Gar04]) und somit gelten die erwähnten Eigenschaften der RKHS-Regularisierung für den Differentialoperator aus (5.10).
Da der Term $||f||_{L_2}$ lediglich Abweichungen der Funktion f vom Koordinatenursprung bestraft – was nicht in unserem Sinn ist, sofern die Funktion nicht zentriert ist –, wird auf diesen verzichtet und lediglich die entsprechende Seminorm $\Phi(f) = |f|_{H^1_{mix}}$ betrachtet. Es sei angemerkt, dass auch ein RKHS $\mathcal{H}$ existiert, welcher der Regularisierung mit der H^1_{mix}-Seminorm zugeordnet werden kann:

$$|f|_{H^1_{mix}} = ||f||_{\mathcal{H}}.$$

In diesem Zusammenhang sind die Theoreme 4.9 und 4.10 in [SS02] interessant. Diese sagen aus, dass es für viele Arten von Differentialoperatoren S möglich ist, einen reproduzierenden Kern mit korrespondierendem RKHS $\mathcal{H}_S$ zu konstruieren. Es gilt dann

$$||S(f)||_{L_2}^2 = ||f||_{\mathcal{H}_S}^2.$$

Diese Konstruktion führt zu Räumen $\mathcal{H}_S$, die als Span verschiedener Eigenvektoren von S^*S entstehen.
Dass eine Betrachtung des H^1-Regularisierungsoperators dennoch sinnvoll ist, wird in [GH09] gezeigt. Dort wird bewiesen, dass die H^1-regularisierte Vorgehensweise ebenfalls als Kern-Methode angesehen werden kann, sofern man einen endlich-dimensionalen Funktionenraum zu Grunde legt. Des Weiteren werden dort auch Fehlerschranken für eine Approximation an die vorliegenden Datenpunkte bewiesen.

5.2 Minimierung des Funktionals

Wir orientieren uns im Folgenden an Abschnitt 3.2 aus [Gar04]. Zu lösen ist das Minimierungsproblem

$$R(f) = \frac{1}{N-d}\sum_{i=d}^{N-1}(s_{i+1} - f(\mathbf{x_i}))^2 + \lambda\Phi(f) \quad \stackrel{f\in\Gamma}{\longrightarrow} \quad \min\,! \tag{5.13}$$

mit $\Phi(f) = |f|^2_\Omega$, wobei $\Omega \in \{H^1, H^1_{mix}\}$.
Es sei nochmals darauf hingewiesen, dass der Raum H^1_{mix} als RKHS – den Raum der möglichen Lösungen direkt bestimmt.

5.2.1 Minimierung in einer beliebigen Basis

Wir nehmen nun an, dass wir durch $\{\gamma_i\}_{i=1}^\infty$ eine Basis von Γ gegeben haben und f sich somit als

$$f(\mathbf{x}) = \sum_{i=1}^\infty \alpha_i\gamma_i(\mathbf{x}),\ \alpha_i \in \mathbb{R}$$

darstellen lässt. In diesem Fall lässt sich $R(f)$ als

$$\begin{aligned} R\left(\sum_{i=1}^\infty \alpha_i\gamma_i\right) &= \frac{1}{N-d}\sum_{j=d}^{N-1}\left(s_{j+1} - \sum_{i=1}^\infty \alpha_i\gamma_i(\mathbf{x_j})\right)^2 + \lambda\left|\sum_{i=1}^\infty \alpha_i\gamma_i\right|^2_\Omega \\ &= \frac{1}{N-d}\sum_{j=d}^{N-1}\left(s_{j+1} - \sum_{i=1}^\infty \alpha_i\gamma_i(\mathbf{x_j})\right)^2 + \lambda\sum_{i=1}^\infty\sum_{j=1}^\infty \alpha_i\alpha_j \sum_{|\mathbf{a}|_*=1}(D^\mathbf{a}\gamma_i, D^\mathbf{a}\gamma_j)_{L_2} \end{aligned}$$

schreiben. Hierbei ist

$$D^\mathbf{a} = \frac{\partial^{a_1}}{\partial x_1^{a_1}}\cdots\frac{\partial^{a_d}}{\partial x_1^{a_d}}.$$

Für die Norm $|\cdot|_*$, in welcher der Multiindex gemessen wird, gilt:

$$\begin{aligned} * = l_1 &\quad\Leftrightarrow\quad \Omega = H^1 \\ * = l_\infty &\quad\Leftrightarrow\quad \Omega = H^1_{mix} \end{aligned}$$

Differenziert man den erhaltenen Ausdruck nach α_k, so erhält man

$$\frac{\partial}{\partial\alpha_k}R(f) = -\frac{2}{N-d}\sum_{j=d}^{N-1}\left(s_{j+1} - \sum_{i=1}^\infty \alpha_i\gamma_i(\mathbf{x_j})\right)\gamma_k(\mathbf{x_j}) + 2\lambda\sum_{j=1}^\infty \alpha_j \sum_{|\mathbf{a}|_*=1}(D^\mathbf{a}\gamma_k, D^\mathbf{a}\gamma_j)_{L_2}.$$

Da $\frac{\partial}{\partial \alpha_k} R(f) = 0 \; \forall \; k = 1, 2, \ldots$ eine notwendige Bedingung für ein Minimum von R ist, erhalten wir die Gleichung

$$\sum_{i=1}^{\infty} \alpha_i \left(\sum_{j=d}^{N-1} \gamma_k(\mathbf{x_j}) \gamma_i(\mathbf{x_j}) + (N-d)\lambda \sum_{|\mathbf{a}|_*=1} \left(D^{\mathbf{a}} \gamma_k, D^{\mathbf{a}} \gamma_i \right)_{L_2} \right) = \sum_{j=d}^{N-1} s_{j+1} \gamma_k(\mathbf{x_j})$$

für alle $k \in \mathbb{N} \setminus \{0\}$. Verwendet man unendlich-dimensionale Vektoren und Matrizen, so kann dies als Gleichungssystem geschrieben werden:

$$\left(\mathbf{B}^T \mathbf{B} + (N-d)\lambda \mathbf{C} \right) \alpha = \mathbf{B}^T \mathbf{s}. \tag{5.14}$$

Hierbei ist $\alpha = (\alpha_1, \alpha_2, \ldots)^T \in \mathbb{R}^{\infty}$ und $\mathbf{s} = (s_{d+1}, \ldots, s_N)^T \in \mathbb{R}^{N-d}$.
$\mathbf{B} \in \mathbb{R}^{(N-d)\times\infty}$ stellt die *Datenmatrix*

$$\mathbf{B}_{i,j} = \gamma_j(\mathbf{x_i})$$

dar und $\mathbf{C} \in \mathbb{R}^{\infty\times\infty}$ ist die *Regularisierungsmatrix*

$$\mathbf{C}_{i,j} = \sum_{|\mathbf{a}|_*=1} \left(D^{\mathbf{a}} \gamma_i, D^{\mathbf{a}} \gamma_j \right)_{L_2}.$$

Es sei angemerkt, dass die Bedingung $\frac{\partial}{\partial \alpha_k} R(f) = 0 \; \forall \; k = 1, 2, \ldots$ auch hinreichend ist, um ein Minimum zu erlangen, da es sich bei $R(f)$ um eine Summe konvexer Funktionen handelt.

5.2.2 Minimierung in Kerndarstellung

Da ein f, welches R bei einer RKHS-Regularisierung minimiert, durch das Repräsenter-Theorem (5.9) dargestellt werden kann, wollen wir nun f mittels dieser Darstellung ausdrücken und erneut die Minimierung des resultierenden Funktionals betrachten. Es sei also

$$f(\mathbf{x}) = \sum_{i=d}^{N-1} \alpha_i k(\mathbf{x_i}, \mathbf{x}), \; \alpha_i \in \mathbb{R},$$

wobei wir o.B.d.A. annehmen, dass die $k(\mathbf{x_i}, \mathbf{x})$ linear unabhängig sind. Andernfalls wählen wir ein linear unabhängiges System maximaler Größe in $\{k(\mathbf{x_i}, \mathbf{x})\}$ und stellen f als Linearkombination bezüglich dieses Systems dar. Anschließend minimieren wir

$$\hat{R}(f) \;:=\; \frac{1}{N-d} \sum_{j=d}^{N-1} \left(s_{j+1} - \sum_{i=d}^{N-1} \alpha_i k(\mathbf{x_i}, \mathbf{x_j}) \right)^2 + \lambda \left\| \sum_{i=d}^{N-1} \alpha_i k(\mathbf{x_i}, \cdot) \right\|_{\mathcal{H}}^2.$$

$$= \frac{1}{N-d}\sum_{j=d}^{N-1}\left(s_{j+1} - \sum_{i=d}^{N-1}\alpha_i k(\mathbf{x_i}, \mathbf{x_j})\right)^2 + \lambda \sum_{i=d}^{N-1}\sum_{j=d}^{N-1}\alpha_i\alpha_j \left\langle k(\mathbf{x_i}, \cdot), k(\mathbf{x_j}, \cdot)\right\rangle_{\mathcal{H}}.$$

Per Definition 5.1 gilt

$$\left\langle k(\mathbf{x_i}, \cdot), k(\mathbf{x_j}, \cdot)\right\rangle_{\mathcal{H}} = k(\mathbf{x_i}, \mathbf{x_j})$$

und die Minimierung von $\hat{R}(f)$ führt zu

$$0 = \frac{\partial}{\partial\alpha_l}\hat{R}(f) = -\frac{2}{N-d}\sum_{j=d}^{N-1}\left(s_{j+1} - \sum_{i=d}^{N-1}\alpha_i k(\mathbf{x_i}, \mathbf{x_j})\right)k(\mathbf{x_j}, \mathbf{x_l}) + 2\lambda\sum_{j=d}^{N-1}\alpha_j k(\mathbf{x_j}, \mathbf{x_l})$$

für alle $l = d, \ldots, N-1$. Somit erhalten wir das Gleichungssystem

$$\sum_{i=d}^{N-1}\alpha_i\left(\sum_{j=d}^{N-1}k(\mathbf{x_i}, \mathbf{x_j})k(\mathbf{x_j}, \mathbf{x_l}) + (N-d)\lambda k(\mathbf{x_i}, \mathbf{x_l})\right) = \sum_{j=d}^{N-1}s_{j+1}k(\mathbf{x_j}, \mathbf{x_l})$$

für alle $l = d, \ldots, N-1$. In Matrixschreibweise ist dies darstellbar als

$$(\mathbf{KK} + (N-d)\lambda\mathbf{K})\,\alpha = \mathbf{Ks}.$$

Hierbei ist $\alpha = (\alpha_d, \ldots, \alpha_{N-1})^T \in \mathbb{R}^{N-d}$ und $\mathbf{s} = (s_{d+1}, \ldots, s_N)^T \in \mathbb{R}^{N-d}$.
$\mathbf{K} \in \mathbb{R}^{(N-d)\times(N-d)}$ stellt die symmetrische *Kernmatrix*

$$\mathbf{K}_{i,j} = k(\mathbf{x_i}, \mathbf{x_j})$$

dar. Aufgrund der Annahme, dass die $k(\mathbf{x_i}, \mathbf{x})$ linear unabhängig sind, ist die Matrix $\mathbf{K}$ strikt positiv definit und das Problem somit eindeutig gestellt. Wir erhalten durch Multiplizieren beider Seiten des Gleichungssystems mit $\mathbf{K}^{-1}$

$$(\mathbf{K} + (N-d)\lambda\mathbf{I})\,\alpha = \mathbf{s}. \tag{5.15}$$

5.3 Dünne Gitter

Um das lineare Gleichungssystem (5.14) numerisch behandeln zu können, ist es notwendig, vom unendlich-dimensionalen Funktionenraum Γ in einen Funktionenraum Γ_m mit endlicher Basis $\{\gamma_i\}_{i=1}^m$ zu wechseln.
In vielen Bereichen des maschinellen Lernens wird eine datenorientierte Basis gewählt. Dies bedeutet, dass sich die Anzahl der Basisfunktionen – und oftmals auch deren Lage im Raum – an den vorliegenden Datenpunkten orientiert. Beispiele für eine solche Basis sind radiale Basisfunktionen oder auch allgemeine datenorientierte Kernfunktionen, wie in der Darstellung (5.15). Diese Methoden sind jedoch nur für eine moderate Anzahl von Datenpunkten geeignet. Wir sehen am Beispiel des Kern-Gleichungssystems, dass es

notwendig ist, die Matrix K aufzustellen, was bei einem d-dimensionalen Problem mit N Zeitreihenpunkten einem Aufwand von $O(d \cdot (N-d)^2)$ entspricht. Ein direktes Lösen des Gleichungssystems – mit einem vollen Aufstellen der Matrix – wäre mit einer Laufzeit von $O(d \cdot (N-d)^2 + (N-d)^3)$ und einer Speicherkomplexität von $O((N-d)^2)$ verbunden. Wie in [Gar04] erwähnt, ist die Lösung dieses Systems, wie sie in Support-Vektor-Maschinen vorgenommen wird, einer direkten Analyse allerdings nicht zugänglich.
Um eine niedrige Kostenkomplexität bezüglich der Anzahl der Datenpunkte zu erhalten, eignen sich raumbasierte Basen. Wir gehen im Folgenden davon aus, dass der Definitionsbereich der γ_i der Einheitswürfel $[0,1]^d$ ist.
Eine leicht zugängliche Methode ist die Überdeckung des Raums mit einem uniformen Gitter der Maschenweite $h_l := \frac{1}{2^l}$ mit $l \in \mathbb{N}$. Jedem Gitterpunkt wird bei diesem Ansatz eine Basisfunktion γ_i zugeordnet. Mit diesem Ansatz beträgt die Anzahl der Freiheitsgrade allerdings $(2^l + 1)^d = O(h_l^{-d})$ und steigt somit exponentiell mit der Dimension d. Dieses Phänomen ist auch als *Fluch der Dimension* bekannt, siehe [Bel57]. Die im Rahmen dieser Arbeit verwendeten *dünnen Gitter* sind jedoch in der Lage den Fluch der Dimension fast komplett zu umgehen und dennoch zum uniformen Gitter vergleichbare Ergebnisse zu liefern.
Wir orientieren uns bei den folgenden Darstellungen vor allem an [FG09, Gar04].

5.3.1 Hierarchische Basen und der Tensorproduktansatz

Volle Gitter und die nodale Basis

Zunächst wählen wir d-lineare Hütchenfunktionen als Basisfunktionen. Ist $d = 1$, so stellt

$$\phi(x) := \begin{cases} 1 - |x|, & \text{falls } x \in [-1,1] \\ 0 & \text{sonst} \end{cases}$$

eine solche Hütchenfunktion dar. Der Träger, der hierbei als $[-1,1]$ gegeben ist, kann durch Translation und Dilatation verändert werden. Wir definieren $\phi_{l,i}$ mit $i, l \in \mathbb{N}$ durch

$$\phi_{l,i}(x) := \phi\left(\frac{x - i \cdot h_l}{h_l}\right)$$

und erhalten somit den Träger $[(i-1)h_l, (i+1)h_l]$, welcher an die Maschenweite $h_l = 2^{-l}$ angepasst ist. Für ein allgemeines d ergibt sich die d-lineare Variante einer Hütchenfunktion durch den folgenden Tensorproduktansatz

$$\phi_{\mathbf{l},\mathbf{i}}(\mathbf{x}) := \prod_{j=1}^{d} \phi_{l_j,i_j}(x_j).$$

Wir nennen $\mathbf{l} = (l_1, \ldots, l_d) \in \mathbb{N}^d$ das multivariate Verfeinerungslevel und $\mathbf{i} = (i_1, \ldots, i_d) \in \mathbb{N}^d$ die multivariate Position der Funktion $\phi_{\mathbf{l},\mathbf{i}}$. Definieren wir nun den Punkt $\mathbf{x}_{\mathbf{l},\mathbf{i}}$ durch

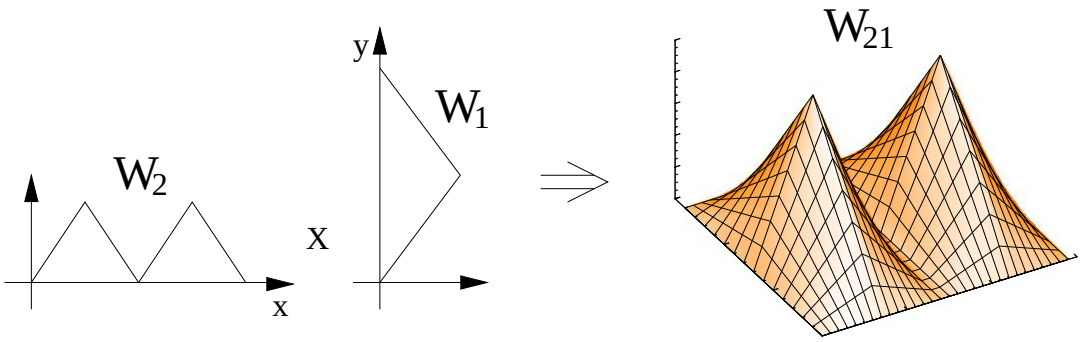


Abb. 5.1: Tensorproduktansatz für stückweise bilineare Funktionen. Abbildung entnommen aus [FG09].

$$\mathbf{x}_{\mathbf{l},\mathbf{i}} := \mathbf{i} \cdot \mathbf{h}_{\mathbf{l}} = (x_{l_1,i_1}, \ldots, x_{l_d,i_d})^T \text{ mit } x_{l_j,i_j} = i_j \cdot h_{l_j},$$

so repräsentiert die Menge $\left\{\mathbf{x}_{\mathbf{l},\mathbf{i}} \mid \mathbf{0} \leq \mathbf{i} \leq 2^{\mathbf{l}}\right\}$ ein Gitter $\Omega_{\mathbf{l}}$, welches in jede Koordinatenrichtung äquidistant ist, aber verschiedene Maschenweiten haben kann. Die Ungleichungen zwischen den Multiindizes sind hierbei und im Folgenden komponentenweise zu verstehen und $\mathbf{0}$ stellt den Nullindex $(0, \ldots, 0)$ dar. Entsprechend erhalten wir den Funktionenraum $V_{\mathbf{l}}$ der stückweise d-linearen Funktionen auf $\Omega_{\mathbf{l}}$ mit Definitionsbereich $[0,1]^d$ durch

$$V_{\mathbf{l}} := \operatorname{span}\left\{\phi_{\mathbf{l},\mathbf{i}|_{[0,1]^d}} \mid \mathbf{0} \leq \mathbf{i} \leq 2^{\mathbf{l}}\right\}.$$

Die Basis der $\phi_{\mathbf{l},\mathbf{i}}$ nennen wir hierbei *nodale* Basis, da in der Darstellung

$$\hat{f}_{\mathbf{l}} := \sum_{\mathbf{i} \leq 2^{\mathbf{l}}} \hat{\alpha}_{\mathbf{l},\mathbf{i}} \phi_{\mathbf{l},\mathbf{i}}$$

einer Funktion aus $V_{\mathbf{l}}$ der Wert an der Stelle $\mathbf{x}_{\mathbf{l},\mathbf{i}}$ dem Wert des Koeffizienten $\hat{\alpha}_{\mathbf{l},\mathbf{i}} \in \mathbb{R}$ entspricht. Wie man sofort sieht, gilt für den Schnitt der Trägers zweier nodaler Basisfunktionen

$$\operatorname{supp}(\phi_{\mathbf{l},\mathbf{i}}) \cap \operatorname{supp}(\phi_{\mathbf{l},\mathbf{j}}) \neq \emptyset \;\Leftrightarrow\; \exists\, \mathbf{k} \in \{-1,0,1\}^d : \mathbf{i} + \mathbf{k} = \mathbf{j}. \tag{5.16}$$

Dünne Gitter und die hierarchische Basis

Der Raum $V_{\mathbf{l}}$ lässt sich auch als eine Summe *hierarchischer Inkremente* darstellen. Hierfür definieren wir zunächst die Indexmenge

$$\mathbf{I}_{\mathbf{l}} := \left\{ \mathbf{i} \in \mathbb{N}^d \;\middle|\; \begin{array}{ll} 0 \leq i_j \leq 1, & \text{falls } l_j = 0 \\ 1 \leq i_j \leq 2^{l_j} - 1, i_j \text{ ungerade} & \text{sonst} \end{array} \quad \forall\, 1 \leq j \leq d \right\}, \tag{5.17}$$

und mit Hilfe dieser die hierarchischen Inkremente

$$W_{\mathbf{l}} := \operatorname{span}\{\phi_{\mathbf{l},\mathbf{i}} \mid \mathbf{i} \in \mathbf{I}_{\mathbf{l}}\}.$$

Aufgrund der Tatsache, dass für alle $j \in \{1, \ldots, d\}$

$$x_{i_j,l_j} = x_{2i_j,l_j+1}$$

gilt, erhalten wir den Zusammenhang

$$W_{\mathbf{l}} = V_{\mathbf{l}} \setminus \bigcup_{i=1}^{d} V_{\mathbf{l}-\mathbf{e}_i},$$

wobei $\mathbf{e}_i$ der i-te Einheitsvektor ist und $V_{\mathbf{l}} = \emptyset$ für alle $\mathbf{l}$ mit $l_j = -1$ für ein $j \in \{1, \ldots, d\}$ gilt. Somit erhalten wir die hierarchische Darstellung

$$V_{\mathbf{l}} = \bigoplus_{\mathbf{k} \leq \mathbf{l}} W_{\mathbf{k}} = \operatorname{span}\{\phi_{\mathbf{k},\mathbf{i}} \mid \mathbf{i} \in \mathbf{I}_{\mathbf{k}}, \mathbf{k} \leq \mathbf{l}\}. \tag{5.18}$$

mit der hierarchischen *Faber*-Basis $\{\phi_{\mathbf{k},\mathbf{i}} \mid \mathbf{i} \in \mathbf{I}_{\mathbf{k}}, \mathbf{k} \leq \mathbf{l}\}$. Die eindimensionale hierarchische Basis ist in Abbildung 5.2 zu sehen. In dieser eindimensionalen Basis gilt nun die

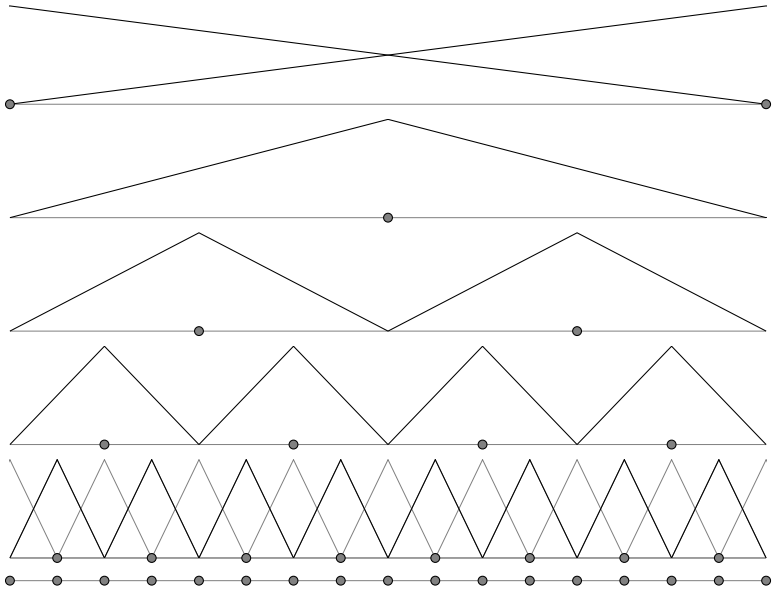


Abb. 5.2: Eindimensionale hierarchische Faber-Basis. Abbildung entnommen aus [FG09].

Beziehung

$$\operatorname{supp}(\phi_{a,i}) \cap \operatorname{supp}(\phi_{b,j}) \neq \emptyset$$

$$\Leftrightarrow \quad \exists\, c \in \{1, 3, \ldots, 2^{a-b+1} - 3, 2^{a-b+1} - 1\} : i = 2^{a-b}(j-1) + c,$$

wenn wir o.B.d.A. annehmen, dass $a > b$ gilt. Im Fall $a = b$ sind die Träger für alle $i \neq j$ disjunkt.
Eine Funktion $f \in V_{\mathbf{l}}$ lässt sich in der d-dimensionalen hierarchischen Basis durch

$$f(\mathbf{x}) = \sum_{\mathbf{k} \leq \mathbf{l}} f_{\mathbf{k}}(\mathbf{x}), \quad f_{\mathbf{k}}(\mathbf{x}) = \sum_{\mathbf{i} \in \mathbf{I_k}} \alpha_{\mathbf{k},\mathbf{i}} \phi_{\mathbf{k},\mathbf{i}}(\mathbf{x})$$

mit $\alpha_{\mathbf{k},\mathbf{i}} \in \mathbb{R}$ darstellen.
Falls $f \in V_{\mathbf{l}}$ der Interpolant einer Funktion $\tilde{f}$ ist, so besagt ein Ergebnis aus [BG04], dass

$$||f_{\mathbf{k}}||_{L_2} = O\left(2^{-2|\mathbf{k}|_1}\right)$$

gilt, sofern $\tilde{f} \in H^2_{mix}$ ist. Zur Definition der Räume und Normen sei auf (5.10), (5.11) und (5.12) hingewiesen. Die Konstante in dieser Abschätzung hängt ausschließlich vom Term

$$||D^{\mathbf{2}} f||_{L_2} = \left|\left| \frac{\partial^{2d}}{\partial x_1^2 \ldots \partial x_d^2} f \right|\right|_{L_2} \tag{5.19}$$

und der Dimension d ab. Eine analoge Abschätzung findet sich auch für $||f_{\mathbf{k}}||_{L_\infty}$, wobei der Term aus (5.19) dann in der L_∞- statt der L_2-Norm zu messen ist.
Dieses Verhalten motiviert die Betrachtung der Räume

$$V_t^s := \bigoplus_{|\mathbf{k}|_1 \leq t} W_{\mathbf{k}},$$

wobei wir $t \in \mathbb{N}$ *Level* nennen.

Definition 5.2 [Dünne Gitter]
Die Menge

$$\Omega_t^s := \{\mathbf{x}_{\mathbf{k},\mathbf{i}} \mid |\mathbf{k}|_1 \leq t, \mathbf{i} \in \mathbf{I_k}\}$$

nennen wir ***reguläres dünnes Gitter*** *zum Level t.*

Für den Interpolanten

$$f_t^s := \sum_{\mathbf{k} \leq t} \sum_{\mathbf{i} \in \mathbf{I_k}} \alpha_{\mathbf{k},\mathbf{i}} \phi_{\mathbf{k},\mathbf{i}} \tag{5.20}$$

in V_t^s lässt sich zeigen, dass für den Interpolationsfehler

$$\left|\left| \tilde{f} - f_t^s \right|\right|_{L_2} = O\left(2^{-2t} \cdot t^{d-1}\right)$$

gilt und die Anzahl der Freiheitsgrade die Gleichung

$$|V_t^s| = O\left(2^t \cdot t^{d-1}\right)$$

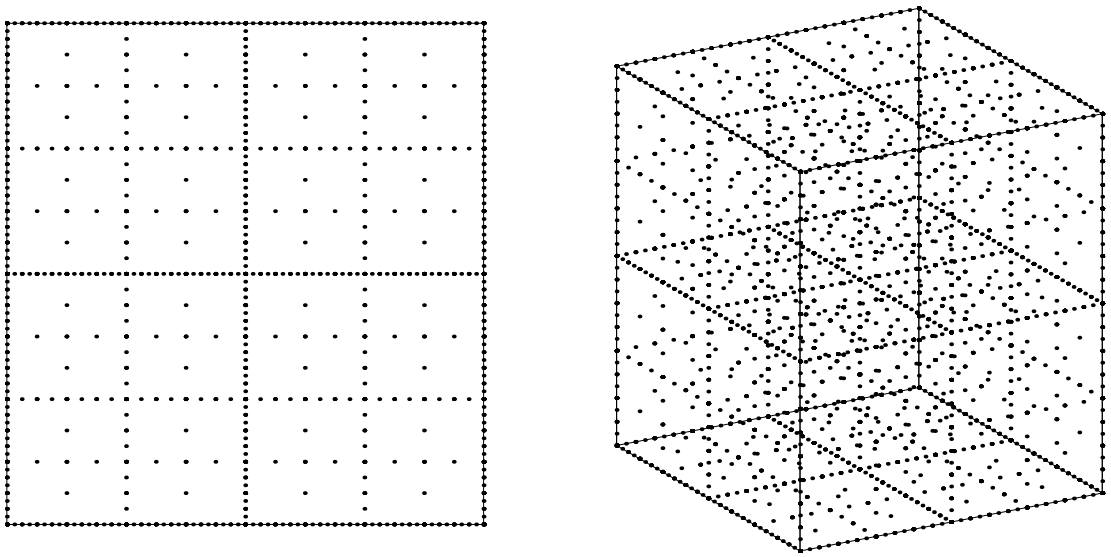


Abb. 5.3: Reguläre dünne Gitter in 2 und 3 Dimensionen. Abbildung entnommen aus [FG09].

erfüllt – siehe auch [Gri06a] für eine detaillierte Betrachtung der Konstante in der Fehlerabschätzung. Der Fluch der Dimension zeigt sich hier in abgeschwächter Form im Exponenten von t. Dies ist ein großer Vorteil im Vergleich zu Approximationen auf den vollen Gittern. Für den entsprechenden Vollgitterraum

$$V_t := \bigoplus_{|\mathbf{k}|_\infty \leq t} W_{\mathbf{k}}$$

beträgt die Anzahl der Freiheitsgrade $O\left(2^{td}\right)$ bei einer geringfügig besseren L_2-Approximationsgenauigkeit von $O\left(2^{-2t}\right)$. Für Beweise der Abschätzungen sei auf [BG04] verwiesen. Es sei noch angemerkt, dass für das Resultat auf vollen Gittern die Forderung $\tilde{f} \in H^2$ ausreicht. Eine analoge Beobachtung gilt für die L_∞-Norm. In der H^1-Seminorm, welche auch als *Energienorm* bezeichnet wird, ist für volle und dünne Gitter eine Approximation mit einem Fehler von $O(2^{-t})$ nachweisbar.
Es ist somit zu sehen, dass mit dünnen Gittern für Funktionen $\tilde{f} \in H^2_{mix}$ trotz einer bemerkenswerten Reduktion der Freiheitsgrade eine ähnlich gute Approximationsgenauigkeit erhalten werden kann wie auf vollen Gittern.
Die Funktion f_t^s kann nun mit der Darstellung aus (5.20) in (5.14) eingesetzt werden, um ein endlich-dimensionales lineares Gleichungssystem zu erhalten, mit welchem die Koeffizienten der Dünngitterlösung berechnet werden können.
Es sei angemerkt, dass eine Zielfunktion g aus (5.2), für welche der zugrundeliegende Prozess die Voraussetzungen von Takens' Theorem 3.1, bzw. 3.2 erfüllt, in H^2 liegt. Dies

ist darauf zurückzuführen, dass der Attraktor von einem Kompaktum umgeben ist und alle verwendeten Funktionen mindestens zweimal stetig differenzierbar sind. Oftmals liegt g aber auch in H^2_{mix}, da die Prozesse und Observablen höheren Glattheitsanforderungen genügen. In den bisher vorgestellten Beispielen der Henon-Abbildung und des Lorenz-Systems waren die betreffenden Funktionen und die bisher verwendeten Observablen beliebig oft differenzierbar.
Bei der Analyse der Zeitreihe sind allerdings der L_2-, bzw. L_∞-Fehler bezüglich des Lebesgue-Maßes auf $[0,1]^d$ nicht von Interesse, da der rekonstruierte Attraktor lediglich einen kleinen Teil dieses Würfels ausfüllt. Die in (5.13) verwendeten Punkte $\mathbf{x_i}$ liegen jedoch – bestenfalls – auf dem Attraktor und es kann somit nicht angenommen werden, dass der L_2-, bzw L_∞-Fehler auf $[0,1]^d$ gegen 0 geht, da in vielen Bereichen des mit dem Gitter überdeckten Würfels keine Datenpunkte liegen. Für unsere Zwecke ist der *Root-Mean-Squared-Error* (RMSE)

$$\sqrt{\frac{1}{M}\sum_{i=1}^{M}(f_t^s(\tilde{\mathbf{x}}_\mathbf{i}) - g(\tilde{\mathbf{x}}_\mathbf{i}))^2}$$

für eine beliebige Auswahl an eingebetteten Punkten $(\tilde{\mathbf{x}}_\mathbf{i})_{i=1}^M$ interessanter, da dieser den L_2-Fehler bezüglich der empirischen Verteilung $\frac{1}{M}\sum_{i=1}^{M}\delta_{\tilde{\mathbf{x}}_\mathbf{i}}$ darstellt, welche nach dem Ergodensatz mit einer Rate von $O\left(\frac{1}{\sqrt{M}}\right)$ gegen die Verteilung μ auf dem Attraktor konvergiert. Der L_2-Fehler bezüglich μ stellt hier ein sinnvolleres Fehlermaß dar als der Fehler bezüglich des Lebesgue-Maßes $\lambda_{[0,1]^d}$. Eine analoge Betrachtung für den L_∞-Fehler führt zum empirischen L_∞-Fehler

$$\max_{i=1,\ldots,M} |f_t^s(\tilde{\mathbf{x}}_\mathbf{i}) - g(\tilde{\mathbf{x}}_\mathbf{i})|.$$

Die $\tilde{\mathbf{x}}_\mathbf{i}$ können hierbei beliebige Punkte auf dem Attraktor darstellen. Um eine Fehleranalyse ohne Bias vorzunehmen, sollten die *Testdaten* $\{\tilde{\mathbf{x}}_\mathbf{i}\}$ jedoch leeren Schnitt mit der Menge der sogenannten *Trainingsdaten* $\{\mathbf{x_i}\}$ haben, welche im Lernfunktional (5.13) verwendet wurden. Zu diesem Thema sei auf [Gar04] verwiesen.

5.3.2 Dünngitter-Kombinationstechnik

Eine andere Methode zur Berechnung einer Dünngitterapproximation stellt die *Kombinationstechnik* dar, siehe auch [GSZ92]. Diese beruht auf dem Prinzip der multivariaten Extrapolation. Hierbei werden Lösungen $\hat{f}_\mathbf{l}$ auf verschiedenen anisotropen vollen Gittern $\Omega_\mathbf{l}$ berechnet und diese zu einer Lösung f_t^c auf dem dünnen Gitter Ω_t^s kombiniert. Die

zugrundeliegende Kombinationsformel ist durch

$$f_t^c(\mathbf{x}) := \sum_{q=0}^{d-1} (-1)^q \begin{pmatrix} d-1 \\ q \end{pmatrix} \sum_{|\mathbf{l}|_1 = t-q} \hat{f}_{\mathbf{l}}(\mathbf{x}) \tag{5.21}$$

gegeben. f_t^c lebt hierbei in V_t^s. Die Vorgehensweise der Kombinationstechnik ist in Abbildung 5.4 anhand eines Beispiels dargestellt. Da wir im Rahmen der Zeitreihenanalyse f_t^c lediglich an verschiedenen Punkten auswerten, ist ein explizites Aufstellen der Funktion durch eine d-lineare Interpolation nicht notwendig, und es genügt die Teillösungen $\hat{f}_{\mathbf{l}}(\mathbf{x})$ zu speichern und bei Bedarf eine Auswertung mittels (5.21) vorzunehmen.

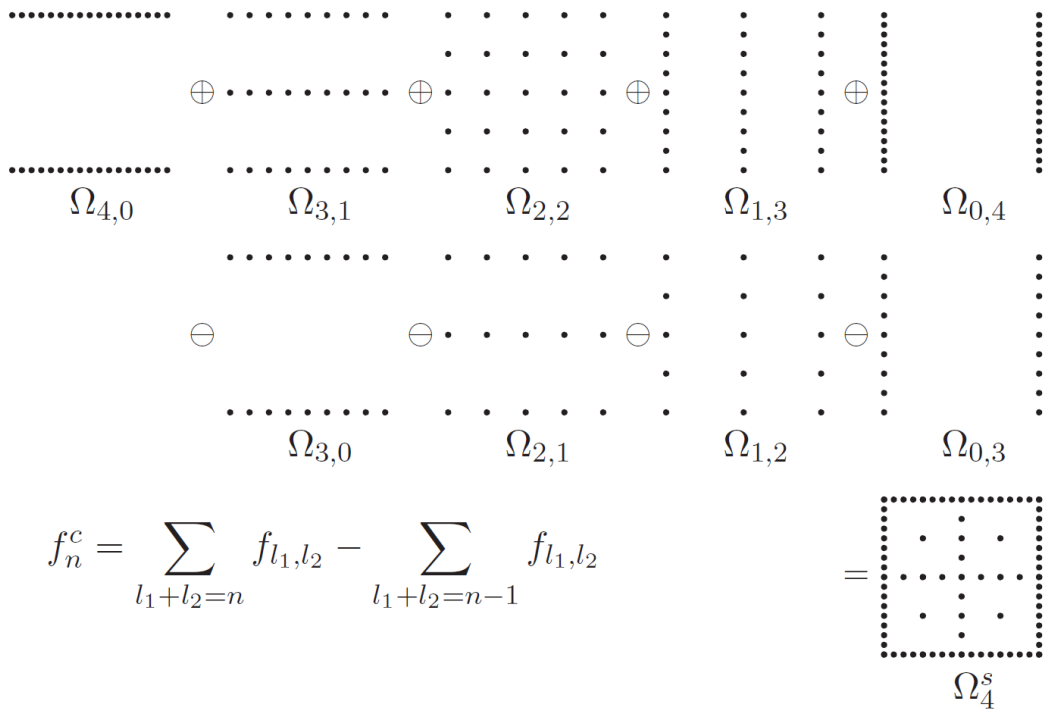


Abb. 5.4: Kombinationstechnik für Level $n = 4$ und Dimension $d = 2$. Abbildung entnommen aus [Gar04].

Bei der Anwendung der Kombinationstechnik werden $O(dt^{d-1})$ Teilprobleme der Größe $O(2^t)$ berechnet. Ein Vorteil gegenüber der direkten Berechnung auf Ω_t^s ist die Tatsache, dass wir nur Ergebnisse auf vollen Gittern berechnen müssen und somit keine hierarchischen Basen benötigen, welche komplizierte Speicher- und Traversierungsmethoden implizieren. Zudem besteht die Möglichkeit einer effizienten Parallelisierung der Berechnungen, siehe [Gar04]. Schlussendlich lässt sich die Kombinationstechnik verwenden, um einen dimensionsadaptiven Dünngitteralgorithmus zu entwerfen, welchen wir in Unter-

abschnitt 5.3.3 vorstellen.
In [GSZ92] wird gezeigt, dass die Kombinationstechnik für Interpolationsprobleme die korrekte Dünngitterlösung f_l^s liefert. In [Gar04] wurde allerdings experimentell gezeigt, dass die Kombinationstechnik bei der Anwendung auf regularisierte Lernverfahren instabil ist und divergieren kann. Aus diesem Grund wurde eine *optimale* Kombinationstechnik entwickelt, welche in unserem Verfahren allerdings nicht verwendet wird. Für Details zur Kombinationstechnik und zur optimalen Kombinationstechnik sei auf [Gar04, HGC07] verwiesen.

5.3.3 Dimensionsadaptive dünne Gitter

Oftmals beinhalten verschiedene Dimensionen der eingebetteten Daten nicht denselben Informationsgehalt. Im Extremfall kann es vorkommen, dass eine Dimension keinerlei relevante Information beinhaltet. In diesen Fällen besteht die Möglichkeit durch veränderte Dünngitter-Räume eine Approximation von g aus (5.2) zu liefern, welche mit weniger Freiheitsgraden (bzw. Gitterpunkten) einen vergleichbaren Approximationsfehler wie ein Ansatz mit regulären dünnen Gittern erzielt.

ANOVA-Zerlegung

Wir wollen eine kurze Motivation für den dimensionsadaptiven Algorithmus geben und stellen deshalb das Konzept der *Analysis of Variance*-Zerlegung (ANOVA-Zerlegung) vor. Für eine ausführliche Auseinandersetzung mit der ANOVA-Zerlegung sei auf [Hol08] verwiesen.
Wir führen eine eindeutige Zerlegung

$$g(x_1, \ldots, x_d) = \sum_{\mathbf{u} \subseteq \{1,\ldots,d\}} g_{\mathbf{u}}(x_{\mathbf{u}}) \tag{5.22}$$

der Funktion g aus (5.2) ein. Hierbei bezeichnen wir mit $x_{\mathbf{u}}$ den $|\mathbf{u}|$-dimensionalen Vektor, der die Komponenten von $\mathbf{x}$ enthält, welche Indizes zugeordnet sind, die in $\mathbf{u}$ liegen. Die Funktionen $g_{\mathbf{u}} : [0,1]^{|\mathbf{u}|} \to \mathbb{R}$ werden im Folgenden rekursiv definiert.
μ bezeichne nun ein beliebiges d-dimensionales Produktmaß auf der Borel-Algebra von $[0,1]^d$:

$$\mu(\mathbf{x}) = \prod_{i=1}^{d} \mu_i(x_i).$$

Eine Funktion g, die im Hilbertraum $\mathcal{H}$ mit Skalarprodukt

$$(f, h) := \int_{[0,1]^d} f(\mathbf{x}) h(\mathbf{x}) d\mu(\mathbf{x})$$

enthalten ist, kann nun wie in (5.22) zerlegt werden, indem wir die entsprechenden Kom-

ponentenfunktionen durch

$$g_{\mathbf{u}}(\mathbf{x}_{\mathbf{u}}) := \sum_{\mathbf{v}\subseteq u}(-1)^{|\mathbf{u}|-|\mathbf{v}|}P_{\mathbf{v}}g(\mathbf{x}_{\mathbf{v}})$$

mit

$$P_{\mathbf{v}}g(\mathbf{x}_{\mathbf{v}}) := \int_{[0,1]^{d-|\mathbf{v}|}} g(x_1,\ldots,x_d)\; d\left(\prod_{i\in\{1,\ldots,d\}\setminus\mathbf{v}}\mu_i(x_i)\right) \tag{5.23}$$

definieren. Es kann gezeigt werden, dass diese Zerlegung eindeutig ist und

$$(g_{\mathbf{u}}, g_{\mathbf{v}}) = 0 \quad \forall\ \mathbf{u}\neq\mathbf{v},\ \ \mathbf{u},\mathbf{v}\neq\emptyset$$

gilt. Aufgrund dieser Orthogonalität lässt sich die Varianz von g durch

$$\sigma^2(g) = \sum_{\mathbf{u}\subseteq\{1,\ldots,d\},\mathbf{u}\neq\emptyset}\sigma^2(g_{\mathbf{u}})$$

ausdrücken. Ist μ nun das d-dimensionale Lebesgue-Maß, so ist $\mathcal{H} = L_2\left([0,1]^d\right)$. Ähnlich zum PCA-Verfahren – siehe (4.26) – kann eine prozentuale Schranke α angegeben werden, welche den Varianzerhalt sicherstellt.

Definition 5.3 [Superpositionsdimension]
*Die **Superpositionsdimension** einer Funktion g ist die kleinste natürliche Zahl d_s, sodass*

$$\sum_{|\mathbf{u}|\leq d_s,\mathbf{u}\neq\emptyset}\sigma^2(g_{\mathbf{u}}) \geq \alpha\sigma^2(g)$$

gilt.

$\alpha = 0.99$ würde bedeuten, dass Korrelationen in mehr als d_s Variablen maximal ein Prozent zur Varianz von g beitragen.
In der Praxis ist die Superpositionsdimension der Zielfunktion g aus (5.2) auch für $\alpha \approx 1$ oftmals kleiner als d. Außerdem kann es vorkommen, dass manche ANOVA-Komponenten einen wesentlich geringeren Beitrag zur Funktion g liefern als andere. Dies zeigt, dass es sinnvoll sein kann, nicht Ω_l^s, sondern andere Gitter zu verwenden, deren Gestalt an die Struktur von g angepasst ist.
[Gar04] beschreibt diesen Effekt mittels verankerter, gewichteter Sobolevräume, welche über das Skalarprodukt

$$\langle f,h\rangle := \sum_{\mathbf{u}\subseteq\{1,\ldots,d\}} w_{\mathbf{u}}^{-1}\int_{[0,1]^{|\mathbf{u}|}}\frac{\partial^{|\mathbf{u}|}}{\partial\mathbf{x}_{\mathbf{u}}}f((\mathbf{x}_{\mathbf{u}},\mathbf{a}_{\{1,\ldots,d\}\setminus\mathbf{u}}))\frac{\partial^{|\mathbf{u}|}}{\partial\mathbf{x}_{\mathbf{u}}}h((\mathbf{x}_{\mathbf{u}},\mathbf{a}_{\{1,\ldots,d\}\setminus\mathbf{u}}))d\mathbf{x}_{\mathbf{u}} \tag{5.24}$$

definiert sind, wobei $w_{\mathbf{u}} \in \mathbb{R}$ eine beliebige Gewichtung darstellt und

$$(\mathbf{x}_{\mathbf{u}}, \mathbf{a}_{\{1,\ldots,d\}\setminus\mathbf{u}})_j := \begin{cases} x_j, & \text{falls } j \in \mathbf{u} \\ a_j & \text{sonst} \end{cases}$$

mit beliebigem, aber festem $\mathbf{a} \in [0,1]^d$ ist. Diese hängen mit der sogenannten *Anker*-ANOVA-Zerlegung zusammen, die man erhält, wenn man die Projektoren nicht wie in (5.23), sondern durch

$$P_{\mathbf{v}} g(\mathbf{x}_{\mathbf{v}}) := g((\mathbf{x}_{\mathbf{v}}, \mathbf{a}_{\{1,\ldots,d\}\setminus\mathbf{v}}))$$

definiert. Es sei angemerkt, dass diese verankerten Sobolevräume ebenfalls Hilberträume mit reproduzierendem Kern sind, siehe [Gar04].
In [Hol08] wird ein direkter Zusammenhang zwischen dünnen Gittern und der ANOVA-Zerlegung für Integrationsprobleme gezeigt. Eine ausführliche Beschreibung des allgemeinen Zusammenhangs zwischen dünnen Gittern und der ANOVA-Zerlegung einer zu approximierenden Funktion findet sich in [Gri06a]. Dort wird die Aufteilung

$$L_2([0,1]) = \mathbf{1} + \mathbf{W}$$

vorgenommen, wobei $\mathbf{1} = \text{span}\{1\}$ der Raum der konstanten Funktionen ist, welchem der Projektor $P_\emptyset$ aus (5.23) zugeordnet werden kann. Diese Idee wird durch einen Tensorproduktansatz

$$L_2^d([0,1]) = L_2([0,1]^d) = \bigotimes_{i=1}^{d} (\mathbf{1}_i + \mathbf{W}_i)$$

verallgemeinert und durch die Wahl einer hierarchischen Faber-Basis (siehe (5.18)) für $\mathbf{W}_i = \mathbf{W}$ erhält man schließlich die entsprechenden Dünngitterräume.
Ähnlich wie der Zusammenhang zwischen gewichteten Sobolevräumen und der Anker-ANOVA-Zerlegung, existiert ein Zusammenhang zwischen der Konstruktion von dünnen Gittern mittels der Hierarchie mit konstanten Funktionen und der Anker-ANOVA-Zerlegung, siehe [Feu10].

Dimensionsadaptivität und die Zulässigkeitsbedingung

Oftmals ist die ANOVA-Struktur einer Funktion nicht bekannt, und es ist nicht möglich, Gitter zu konstruieren, die an die zu lernende Funktion angepasst sind. Stattdessen kann aber ein dimensionsadaptiver Zugang verwendet werden, um ein solches Gitter während der Laufzeit zu generieren. Hierzu definieren wir zunächst verallgemeinerte dünne Gitter über eine sogenannte *Zulässigkeitsbedingung*, siehe auch [GG03].

Definition 5.4 [Zulässigkeitsbedingung]
Eine Indexmenge $\mathbf{I} \subset \mathbb{N}^d$ *heißt* ***zulässig****, falls für jeden Index* $\mathbf{i} \in \mathbf{I}$

$$\mathbf{j} \leq \mathbf{i} \Rightarrow \mathbf{j} \in \mathbf{I}$$

gilt.

Es ist direkt ersichtlich, dass die Indexmengen $\{\mathbf{i} \mid |\mathbf{i}|_1 \leq t\}$, welche reguläre dünne Gitter definieren, zulässig sind. Wir definieren nun verallgemeinerte dünne Gitter:

Definition 5.5 [Verallgemeinerte dünne Gitter]
Sei $\mathbf{I}$ *eine zulässige Indexmenge, dann ist der Raum*

$$V_{\mathbf{I}} := \bigoplus_{\mathbf{i} \in \mathbf{I}} W_{\mathbf{i}}$$

ein ***verallgemeiner Dünngitterraum****. Die korrespondierenden verallgemeinerten dünnen Gitter sind durch*

$$\Omega_{\mathbf{I}}^s := \{\mathbf{x}_{\mathbf{k},\mathbf{i}} \mid \mathbf{k} \in \mathbf{I}, \mathbf{i} \in \mathbf{I}_{\mathbf{k}}\}$$

definiert.

Es sei angemerkt, dass beispielsweise die Indexmenge $\{\mathbf{i} \mid |\mathbf{i}|_\infty \leq t\}$ ebenfalls zulässig ist, das korrespondierende verallgemeinerte dünne Gitter allerdings ein reguläres volles Gitter ist. Somit ist die Bezeichnung hier ein wenig irreführend.
Die Zulässigkeitsbedingung garantiert in diesem Fall, dass sämtliche hierarchischen Vorfahren eines Gitterpunktes $\mathbf{x}_{\mathbf{l},\mathbf{i}}$ existieren. Die direkten hierarchischen Vorfahren für komponentenweise ungerade $\mathbf{i}$ können durch

$$\mathrm{HierAnc}(\mathbf{x}_{\mathbf{l},\mathbf{i}}) := \{\mathbf{x}_{\mathbf{l},\mathbf{j}} \in \Omega_{\mathbf{l}} \mid \exists\, k \in \{1, \ldots, d\}, l_k > 0 : \mathbf{j} = \mathbf{i} \pm \mathbf{e}_k\}$$

definiert werden, wobei $\mathbf{e}_k$ den k-ten Einheitsvektor darstellt. Sind die direkten hierarchischen Vorfahren jedes Punktes im Gitter enthalten, so sind sämtliche hierarchischen Vorfahren aller Punkte enthalten.
Die Kombinationsformel für verallgemeinerte dünne Gitter kann durch

$$f_{\mathbf{I}}^c(\mathbf{x}) := \sum_{\mathbf{i} \in \mathbf{I}} \left(\sum_{\mathbf{k}=\mathbf{0}}^{\mathbf{1}} (-1)^{|\mathbf{k}|_1} \chi^{\mathbf{I}}(\mathbf{i} + \mathbf{k}) \right) \hat{f}_{\mathbf{i}}(\mathbf{x}) \tag{5.25}$$

angegeben werden, wobei

$$\chi^{\mathbf{I}}(\mathbf{j}) := \begin{cases} 1, & \text{falls } \mathbf{j} \in \mathbf{I} \\ 0 & \text{sonst} \end{cases}$$

die Indikatorfunktion der Indexmenge $\mathbf{I}$ darstellt, siehe auch [Gar04].
In [Heg03] wird die Anwendung eines dimensionsadaptiven Dünngitter-Algorithmus auf Interpolationsprobleme vorgestellt. Der in dieser Arbeit verwendete dimensionsadaptive Dünngitter-Algorithmus ist derselbe, der in [Gar04] verwendet wurde und basiert auf [GG03], wo eine Übertragung des Ansatzes von Hegland auf Integrationsprobleme beschrieben wird.
Der Algorithmus soll während der Laufzeit eine zulässige Indexmenge $\mathbf{I}$ konstruieren,

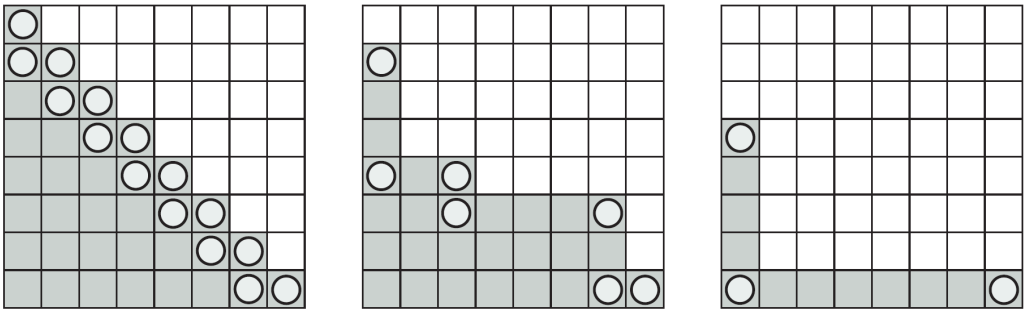


Abb. 5.5: Kombinationstechnik für verallgemeinerte dünne Gitter in $d = 2$ für drei verschiedene Indexmengen $\mathbf{I}$ – die x-Achse stellt hierbei l_1, die y-Achse l_2 dar. Zu sehen sind die Tupel $\{\mathbf{l} = (l_1, l_2) \in \mathbb{N}^2 \mid |\mathbf{l}|_\infty \leq 7\}$ Die Elemente $\mathbf{l} = (l_1, l_2) \in \mathbf{I}$ sind grau unterlegt. Ein Kreis kennzeichnet die Gitter, welche in der Formel der Kombinationstechnik (5.25) einen Koeffizienten haben, der nicht Null ist. Die Abbildung links zeigt die Indexmenge des regulären dünnen Gitters Ω_7^s, die anderen beiden zeigen allgemeine Indexmengen. Abbildung entnommen aus [Gar04].

welche zu einem verallgemeinerten dünnen Gitter korrespondiert, das möglichst gut an die ANOVA-Struktur der Funktion g angepasst ist. Das Verfahren beginnt mit dem kleinsten Index $I := \{\mathbf{0}\}$ und fügt iterativ jeweils einen neuen Index $\mathbf{j}$ zu $\mathbf{I}$ hinzu, sofern

- $\mathbf{I} \cup \{\mathbf{j}\}$ weiterhin zulässig ist und
- das Teilproblem $\hat{f}_\mathbf{j}$ einen möglichst großen Beitrag zur Lösung des Gesamtproblems liefert.

Um den zweiten Punkt zu erfüllen, ist ein lokaler Fehlerindikator nötig, welcher aufgrund einer Heuristik entscheidet, wie der Beitrag des Teilproblems $f_\mathbf{j}^s$ zu gewichten ist. Um widerum den Approximationsfehler der Gesamtlösung zu schätzen, ist ein globaler Fehlerindikator notwendig, welcher ein Kriterium angibt, wann der Algorithmus terminiert und die endgültige Indexmenge $\mathbf{I}$ liefert.
Wie bereits zu erahnen ist, ist es möglich, dass dieser Algorithmus terminiert, bevor eine adäquate Indexmenge gefunden wurde, welche die Struktur von g widerspiegelt. Da lediglich Indizes zu $\mathbf{I}$ hinzugefügt werden, die die Zulässigkeitsbedingung nicht verletzen, kann es vorkommen, dass die lokalen Fehlerindikatoren kleine Beiträge der Indizes signalisieren, die im nächsten Schritt hinzugefügt werden können, obwohl ein Index $\mathbf{j}$ existiert, für welchen $\mathbf{I} \cup \{\mathbf{j}\}$ nicht zulässig ist, $\hat{f}_\mathbf{j}$ allerdings einen großen Beitrag zur Gesamtlösung liefert.
Die genaue Vorgehensweise des Algorithmus sowie die Gestalt möglicher Fehlerindikatoren ist in Unterabschnitt 6.3.2 zu sehen.

Es sei erwähnt, dass in [Feu10] eine Variante eines dimensionsadaptiven Dünngitteralgorithmus zu finden ist, welche direkt auf dem dünnen Gitter arbeitet. Dies erfordert effiziente Speicherstrukturen, die es ermöglichen das dünne Gitter zu traversieren. Die Vorgehensweise ist dennoch ähnlich zum hier angeführten Algorithmus. Allerdings werden fehlende hierarchische Vorfahren nachträglich eingefügt, wodurch sich die Möglichkeit ergibt ANOVA-Strukturen zu erkennen, welche mit dem hier verwendeten Algorithmus nicht adaptiv approximiert werden können. Hält man sich an die restriktive Zulässigkeitsbedingung, so kann ein Teilgitter, für welches der Fehlerindikator einen großen Beitrag zur Lösung signalisiert, nicht hinzugefügt werden, sofern nicht sämtliche hierarchischen Vorfahren vorhanden sind.

Hierarchie mit konstanten Funktionen

Es ist möglich, eine geringfügig andere hierarchische Konstruktion der Räume $V_{\mathbf{I}}$ vorzunehmen, indem man die Indexmengen $\mathbf{I_l}$ aus (5.17) als

$$\tilde{\mathbf{I}}_{\mathbf{l}} := \left\{ \mathbf{i} \in \mathbb{N}^d \;\middle|\; \begin{array}{ll} i_j = 0, & \text{falls } l_j = -1, 0 \\ 1 \leq i_j \leq 2^{l_j} - 1, i_j \text{ ungerade} & \text{sonst} \end{array} \quad \forall\, 1 \leq j \leq d \right\}$$

definiert und die Levelindizes nun Werte aus $(\mathbb{N} \cup \{-1\})^d$ annehmen können. Des Weiteren definiert man neue eindimensionale Ansatzfunktionen $\tilde{\phi}_{l,j}$ durch

$$\begin{aligned} \tilde{\phi}_{-1,0} &:= 1, \\ \tilde{\phi}_{0,0} &:= \phi_{0,1}, \\ \tilde{\phi}_{l,j} &:= \phi_{l,j} \quad \text{für } l \geq 1. \end{aligned}$$

Die entsprechenden Tensorprodukte bilden die neue hierarchische Basis. Da $\tilde{\phi}_{-1,0} - \tilde{\phi_{0,0}} = \phi_{0,0}$ ist, wurde der Span der Basis nicht verändert. Die neuen hierarchischen Teilräume sind darstellbar durch

$$\tilde{W}_{\mathbf{l}} = \operatorname{span}\{\tilde{\phi}_{\mathbf{l},\mathbf{i}} \mid \mathbf{i} \in \tilde{\mathbf{I}}_{\mathbf{l}}\}.$$

Man erhält somit

$$V_{\mathbf{I}} = \bigoplus_{\mathbf{l} \in \mathbf{I}} \tilde{W}_{\mathbf{l}}$$

in der neuen Basisdarstellung. Die Formel (5.21) der Kombinationstechnik bleibt erhalten, doch auch dort wird nun das Level -1 zugelassen.
Der Unterschied zur bisher verwendeten hierarchischen Basis liegt darin, dass eine $(k-1)$-lineare Basisfunktion durch die Verknüpfung mit einer Konstanten keine zusätzlichen Freiheitsgrade erhält und somit auch die verknüpfte Funktion $(k-1)$-linear ist. Das kleinste Teilgitter $\Omega_{\mathbf{0}}$ der alten Hierarchie hatte 2^d Freiheitsgrade, wohingegen das kleinste Teilgitter der neuen Hierarchie $\Omega_{-\mathbf{1}}$ ist und nur noch einen Freiheitsgrad hat. Diese Reduktion ist für reguläre Berechnungen mit der Kombinationstechnik irrelevant, da dort die Gesamtkomplexität gleich bleibt. Für den dimensionsadaptiven Algorithmus hat eine

solche Darstellung allerdings Vorteile, da die Ordnung der erhaltenen Gitter nun nicht länger von der Raumdimension d abhängt, sondern bestenfalls nur noch von der Superpositionsdimension d_s der Funktion g. Dies ist der Fall, wenn der lokale Fehlerindikator in Dimensionen, die keinen signifikanten Beitrag zur Varianz von g liefern, in der Lage ist diese als irrelevant einzustufen und somit in diesen Richtungen nicht verfeinert werden muss.

5.3.4 Ortsadaptive dünne Gitter

Eine dimensionsadaptive Herangehensweise ist sinnvoll, falls wir g im gesamten Würfel rekonstruieren wollen und falls die vorliegenden Datenpunkte hinreichend gut in diesem Würfel verteilt sind. Es wurde bereits erwähnt, dass der Attraktor (auch nach der Skalierung auf $[0,1]^d$) im Allgemeinen lediglich einen kleinen Teil des Würfels ausfüllt und somit nur dort Datenpunkte zu erwarten sind. Außerdem ist es in der Praxis oftmals der Fall, dass unglatte oder sogar unstetige Prozesse vorliegen. Es ist somit nicht nur sinnvoll verschiedene Koordinatenrichtungen unterschiedlich aufzulösen, sondern auch eine *ortsadaptive* Variante des Algorithmus zu betrachten, die es ermöglicht, die Gitter lediglich lokal zu verfeinern, um den Attraktor und vorhandene Unstetigkeitsstellen hinreichend genau aufzulösen. Die Hoffnung ist auch hier, dass dieser Ansatz den gleichen Approximationsfehler wie der Ansatz mit regulären dünnen Gittern liefert, aber weniger Freiheitsgrade benötigt.
Dieser Ansatz wird im Rahmen dieser Arbeit ohne die Kombinationstechnik realisiert, was zur Folge hat, dass die Lösung direkt auf dem dünnen Gitter berechnet werden muss. Ähnlich wie im dimensionsadaptiven Fall muss auch hier sichergestellt werden, dass die resultierenden dünnen Gitter eine Struktur haben, die garantiert, dass keine *hängenden Knoten* entstehen und sämtliche hierarchischen Vorfahren eines Gitterpunktes ebenfalls in der Menge der Gitterpunkte enthalten sind. Dies geschieht in der Weise, dass beim Hinzufügen eines neuen Punktes alle fehlenden Vorfahren ebenfalls in das Gitter eingefügt werden, siehe [Feu10].
Die Anfangskonfiguration ist hierbei durch ein reguläres dünnes Gitter Ω_L^s auf einem niedrigen Level $L = t$ gegeben. Nun wird für jeden Gitterpunkt ein vorgegebener Fehlerindikator ausgewertet. Falls dieser signalisiert, dass eine Verfeinerung notwendig ist, werden sämtliche $2d$ *Kindknoten* in das Gitter eingefügt. Falls notwendig, werden ebenfalls fehlende hierarchische Vorfahren eingefügt. Stellen wir nun sämtliche existierenden Punkte bezüglich des Levels $\mathbf{L} := (L, \ldots, L)$ dar, so sind die Kindknoten zu einem Gitterpunkt $\mathbf{x_{L,i}}$ gerade die Punkte

$$\{\mathbf{x_{L+1,j}} \mid \exists\, k \in \{0, \ldots, d\} : \mathbf{j} \pm \mathbf{e}_k = 2 \cdot \mathbf{i}\}\,,$$

wobei $\mathbf{e}_k$ der k-te Einheitsvektor ist. Wurden alle Punkte durchlaufen, so wird $L := L+1$ gesetzt und der Vorgang auf dem neuen Gitter wiederholt. Dies wird solange iteriert, bis der Fehlerindikator bei keinem Gitterpunkt mehr signalisiert, dass eine Verfeinerung

notwendig ist oder bis ein gewisses $L = L_{max}$ erreicht wurde.
Auch hier ist es offensichtlich, dass der Algorithmus – abhängig vom Fehlerindikator – terminieren kann, bevor eine zufriedenstellende Approximation erreicht wurde.

6 Implementierung und Laufzeitanalyse

In diesem Kapitel sollen die im Rahmen dieser Arbeit erstellten und verwendeten Implementierungen der Dimensions- und Delay-Schätzer sowie der Dünngitter-Vorhersage-Methoden erläutert werden. Insbesondere wird die Komplexität der Verfahren in Bezug auf die Anzahl der vorliegenden Zeitreihenpunkte N sowie die Einbettungsdimension d untersucht. Wir nehmen an, dass jegliche Zeitreihen auf $[0, 1]$ skaliert sind.

6.1 Dimensionsschätzer

6.1.1 Boxcounting-Schätzer

Wie bereits in 4.2 ausgeführt wurde, wird beim Boxcounting-Schätzer die zu approximierende Größe

$$d_{cap}(X) = \lim_{\epsilon \searrow 0} \frac{\log Z(\epsilon)}{-\log \epsilon}$$

durch einen Finite-Differenzen-Ansatz für kleine Parameter $\max C > \epsilon_1 > \epsilon_2 > \min C$ mit

$$C := \{||\mathbf{x_i} - \mathbf{x_j}||_\infty \mid \mathbf{x_i} \neq \mathbf{x_j};\ i, j \in \{d, \ldots, N-1\}\} \tag{6.1}$$

approximiert. Unser Code berechnet also

$$\hat{d}_{cap}(X)(\epsilon_1, \epsilon_2) := \frac{\log Z(\epsilon_1) - \log Z(\epsilon_2)}{\log \epsilon_1 - \log \epsilon_2} = \frac{\log \frac{Z(\epsilon_1)}{Z(\epsilon_2)}}{\log \frac{\epsilon_1}{\epsilon_2}}.$$

Insbesondere im Nenner kann es beim Ausrechnen des Ausdrucks auf der linken Seite zu numerischer Auslöschung kommen. Der Grund hierfür ist die Tatsache, dass – für kleine ϵ_1, ϵ_2 – Zahlen großen Betrags voneinander subtrahiert werden. Daher berechnen wir in unserem Algorithmus den rechten Term.
Die naive Implementierung der Idee, den gesamten Würfel $[0, 1]^d$ mit Boxen der Länge ϵ_1 und ϵ_2 zu überdecken und diese zu durchlaufen, erreicht schnell die maximale Speicherkapazität eines Rechners, da wir somit im d-dimensionalen Raum $O(\lceil \frac{1}{\epsilon_1} \rceil^d + \lceil \frac{1}{\epsilon_2} \rceil^d) = O(\lceil \frac{1}{\epsilon_2} \rceil^d)$ Boxen benötigen. Dies führt bereits in Dimension 3 für den realistischen Fall $\epsilon_2 = 10^{-5}$ zu einer Billiarde Boxen, welche abgespeichert und durchlaufen werden müssen. Geht man davon aus, dass jede Box eine Information von einem Byte speichert, so führt dieser Ansatz zu einer Speicherbelegung von 1000 Terrabyte. Um diese exponentielle Abhängigkeit der Komplexität von der Dimension zu umgehen, wurde eine andere

Vorgehensweise implementiert.
Die Idee des im Rahmen dieser Arbeit implementierten Algorithmus ist, lediglich Boxen abzuspeichern, die auch mit Punkten besetzt sind. Dies geschieht in ähnlicher Weise wie in [LT89, Kru96]. $Z(\epsilon)$ wird dabei berechnet, indem jedem der $(N-d)$ Datenpunkte seine korrespondierende Box anhand eines d-dimensionalen Index zugeordnet wird. Dies geschieht in $O(d \cdot (N-d))$ Operationen. Dann wird die Anzahl der verschiedenen existierenden Indizes gezählt und man erhält somit die Gesamtzahl der okkupierten Boxen. Um nicht alle möglichen $\lceil \frac{1}{\epsilon} \rceil^d$ Indizes überprüfen zu müssen – und wieder den Fluch der Dimension in der Kostenkomplexität zu erhalten –, werden diese vorher lexikografisch angeordnet. Zum Schluss muss die gesamte Indexliste noch einmal durchlaufen werden, um die Anzahl verschiedener Indizes zu zählen. Dies kann auch wieder in $O(d \cdot (N-d))$ geschehen.
Der Term, welcher die Kosten dominiert, ist hierbei durch das lexikografische Sortieren vorgegeben. Hierzu wurde ein gewöhnlicher Quicksort-Algorithmus verwendet, wobei die Ordnung auf der Indexmenge durch

$$\mathbf{i} < \mathbf{j} \Leftrightarrow \exists\ k \in \{1, \ldots, d\} : \mathbf{i}_k < \mathbf{j}_k \text{ und } \mathbf{i}_l = \mathbf{j}_l \forall\ l = 1, \ldots, k-1$$

gegeben ist. Gilt $\mathbf{i} \not< \mathbf{j}$ und $\mathbf{j} \not< \mathbf{i}$, so ist $\mathbf{i} = \mathbf{j}$. Die Laufzeit des d-dimensionalen Quicksort-Algorithmus ist im schlechtesten Fall $O(d \cdot (N-d)^2)$. Die *Average-Case*-Analyse liefert eine Laufzeit von

$$O(d \cdot (N-d) \log_2(N-d)),$$

was die Laufzeit des Schätzers dominiert.
Der Speicherbedarf ist hierbei $O(d \cdot (N-d))$ für das Abspeichern der Punkte und der Box-Indizes. Quicksort benötigt zusätzlich höchstens $O(d \cdot (N-d))$ Speicher für rekursive Aufrufe. Somit liegt der gesamte Speicherbedarf bei

$$O(d \cdot (N-d)).$$

Für eine gute Schätzung der Boxcounting-Dimension ist die richtige Wahl von ϵ_1 und ϵ_2 essentiell. Da es a priori schwierig ist, diese zu bestimmen, wird analog zu [LV07, KS04] vorgegangen. Wir setzen

$$\epsilon_{max} := \frac{\max C}{S} \quad \text{und} \quad \epsilon_{min} := \frac{\max C}{s},\ s > S \in \mathbb{N}$$

mit C aus Gleichung (6.1). Anschließend wird eine Anzahl $A \in \mathbb{N}$ vorgegeben und

$$\epsilon_i := \epsilon_{max} \cdot \left(\frac{\epsilon_{min}}{\epsilon_{max}} \right)^{\frac{i}{A}} \forall\ i = \{0, \ldots, A\} \tag{6.2}$$

gesetzt, da wir wollen, dass die ϵ_i auf der logarithmischen Skala äquidistant sind.

Algorithmus 1 Berechnung von $Z(\epsilon)$

Eingabe: $\epsilon \in (0,1]$, $\mathbf{X} = (\mathbf{x_i})_{i=d}^{N-1}, \mathbf{x_i} \in [0,1]^d \ \forall i$
Ausgabe: $Z(\epsilon)$: die Anzahl belegter Boxen in einem ϵ-Gitter

```
m ← ⌈1/ε⌉      // m wird auf Anzahl Boxen/Dimension gesetzt
for i = d, ..., N − 1 do
        I(i) ← ⌊m · x_i⌋      // Indexbestimmung (elementweises Abrunden)
end for
Quicksort(X)      // Sortiere X lexikografisch nach Indizes
Z(ε) ← 0      // Setze den Boxcounter auf 0
ι ← I(d)      // Setze ι auf den ersten Index
for i = d + 1, ..., N − 1 do
        if ι ≠ I(i) then
                ι ← I(i)      // Setze ι auf den aktuellen Index
                Z(ε) ← Z(ε) + 1      // Inkrementierung des Boxcounters
        end if
end for
return Z(ε)      // Anzahl okkupierter Boxen zurückgeben
```

Schlussendlich werden die Approximationen

$$d_{cap}^{\epsilon_i} := \hat{d}_{cap}(X)(\epsilon_{i-1}, \epsilon_i) \ \forall \ i = \{1, \ldots, A\} \tag{6.3}$$

berechnet. Trägt man nun $d_{cap}^{\epsilon_i}$ gegen $\log \epsilon_i$ auf, wäre es optimal eine horizontale Linie zu erhalten, da dies bedeuten würde, dass man für alle Paare $(\epsilon_{i-1}, \epsilon_i)$ denselben Wert für $d_{cap}^{\epsilon_i}$ errechnet hat. Oftmals ist es jedoch so, dass der Wert von $d_{cap}^{\epsilon_i}$ stark von ϵ_i abhängt. Um eine zuverlässige Schätzung für d_{cap} zu erhalten, müssen nun mehrere aufeinanderfolgende Indizes $i = j, \ldots, j + l$ gesucht werden, in welchem die Werte $d_{cap}^{\epsilon_i}$ nahezu identisch sind (*Plateau*). Das Mittel dieser $d_{cap}^{\epsilon_i}$ liefert eine gute Schätzung für die Boxcounting-Dimension.
Diese Vorgehensweise führt zu einer durschnittlichen Kostenkomplexität von

$$O(A \cdot d \cdot (N-d) \log_2(N-d)).$$

Der Speicherbedarf ändert sich nicht, da maximal zwei der $Z(\epsilon_i)$ gleichzeitig gespeichert sein müssen.
Es scheint, dass man den Fluch der Dimension komplett umgangen hat, jedoch haben wir in Gleichung (4.20) gesehen, dass für eine verlässliche Schätzung

$$d_{cap} < 2 \log_{10}(N-d)$$

gefordert wird und der Fluch der Dimension somit zumindest in der *intrinsichen* Dimen-

sion d_{cap} auftritt. Es bleibt die Hoffnung, dass wir in den meisten praxisrelevanten Fällen mit weniger Datenpunkten auskommen (siehe [Eng91]).

6.1.2 Korrelationsdimensionsschätzer

Zum Schätzen der Korrelationsdimension muss der Term

$$C(\epsilon, n_{\min}) = \left(\frac{2}{(N-d-n_{\min})(N-d-n_{\min}-1)} \cdot \sum_{i=d}^{N-1} \sum_{j=i+1+n_{\min}}^{N-1} H\left(\epsilon - ||\mathbf{x_i} - \mathbf{x_j}||_2\right) \right)$$

aus Gleichung (4.22) ausgewertet werden. Eine naive Implementation, in welcher für alle Punkte der Abstand zu allen anderen Punkten berechnet wird, benötigt $O(d \cdot (N-d)^2)$ viele Operationen. Diese bietet sich jedoch nur für eine moderate Anzahl an Zeitreihenpunkten N an. Im Rahmen dieser Arbeit wurde ein Algorithmus implementiert, welcher sich an [The87] orientiert und die Idee nutzt, die bereits beim Boxcounting-Schätzer verwendet wurde. Im Rahmen der Laufzeit-Abschätzungen sowie unseren Experimenten im folgenden Kapitel werden wir den Term n_{min} der Theiler-Korrektur vernachlässigen, da dieser in der Praxis ohnehin $O(1)$ ist und bei den Berechnungen der Experimente zu keinen verbesserten Ergebnissen geführt hat.
Wie beim Boxcounting-Schätzer wird jedem Datenpunkt $\mathbf{x_i}$ ein d-dimensionaler Index $\mathbf{k_i}$ zugeordnet, welcher die korrespondierende Box $B_{\mathbf{k_i}}$ in einem ϵ-Gitter darstellt. Die Idee ist nun, dass zur Auswertung der Heavyside-Funktion $H\left(\epsilon - ||\mathbf{x_i} - \mathbf{x_j}||_2\right)$, lediglich die Punkte $\mathbf{x_j}$ relevant sind, die in einer Box $B_{\mathbf{k_j}}$ mit

$$|\mathbf{k_j} - \mathbf{k_i}|_\infty \leq 1.$$

liegen. Dies liegt an der Tatsache, dass

$$||\mathbf{x_i} - \mathbf{x_j}||_2 \leq ||\mathbf{x_i} - \mathbf{x_j}||_\infty$$

gilt.
Mittels eines Quicksort-Algorithmus werden die Punkte anhand ihrer d-dimensionalen Indizes sortiert. Danach wird über alle Punkte $\mathbf{x_i}$ iteriert und es werden jeweils die $2 \cdot 3^{d-1}$ benachbarten[1] Boxen gesucht, welche Punkte enthalten können, die noch nicht durchlaufen wurden. Für jeden Punkt, der noch nicht durchlaufen wurde und sich in einer dieser Boxen befindet, wird die Heavyside-Funktion ausgewertet. Die relevanten Boxen sind hierbei

$$B_{\mathbf{k_j}} \quad \text{mit} \quad |\mathbf{k_j} - \mathbf{k_i}|_\infty \leq 1 \quad \text{und} \quad \mathbf{k_j^1} \geq \mathbf{k_i^1}.$$

Der Suchvorgang wird in Algorithmus 2 dargestellt.
Das Zuordnen und Sortieren der Indizes hat hierbei dieselbe Komplexität wie beim Boxcounting-Schätzer. Das Auswerten der Heavyside-Funktion benötigt $O(d)$ viele Ope-

[1] Wir bezeichnen hierbei auch die Box, welche $\mathbf{x_i}$ enthält, als "benachbart".

rationen. Sind die benachbarten Boxen eines $\mathbf{x_i}$ bekannt, so beträgt die Laufzeit für alle Auswertungen der Heavyside-Funktion mit relevanten Punkten aus benachbarten Boxen $O(d \cdot (n_1 + n_2 + \cdots + n_{2 \cdot 3^{d-1}}))$, wobei n_i die Anzahl der Elemente in der i-ten Nachbarbox sind. Dies kann von oben durch $O(\min(d \cdot 2 \cdot 3^{d-1} \cdot n_{max}, (N-d)))$ abgeschätzt werden, wobei n_{max} die Anzahl der Elemente in der Box ist, welche am meisten Punkte enthält. Das Suchen der benachbarten Boxen wird mittels eines Binary-Search-Algorithmus realisiert. Die Laufzeit für einen Suchvorgang kann von oben durch $O(d \cdot \log_2(N-d))$ abgeschätzt werden. Für jeden Punkt tätigt unser Algorithmus $2 \cdot 3^{d-1}$ Binary-Search-Vorgänge.
Somit lässt sich der Gesamtaufwand[2] für die Berechnung von $C(\epsilon)$ durch

$$\begin{aligned} & \underbrace{(N-d)}_{\text{Anzahl Iterationen}} \quad O(\underbrace{d \cdot \min(2 \cdot 3^{d-1} \cdot n_{max}, (N-d))}_{\text{Auswerten}} + \underbrace{d \cdot 2 \cdot 3^{d-1} \cdot \log_2(N-d)}_{\text{Suchen}}) \\ = \quad & O(d \cdot (N-d) \cdot ((N-d) + 2 \cdot 3^{d-1} \log_2(N-d))) \end{aligned}$$

von oben abschätzen. Im schlimmsten Fall sind wir somit schlechter, als der naive Algorithmus mit $O(d \cdot (N-d)^2)$. Wie oben beschrieben sind diese Schranken allerdings lediglich Worst-Case-Abschätzungen. In der Praxis ist meist die Abschätzung durch $O(2 \cdot 3^{d-1} \cdot d \cdot (N-d) \log_2(N-d))$ möglich, da n_{max} genügend klein ist. Für eine detailliertere Auseinandersetzung mit Laufzeit-Abschätzungen für diesen Algorithmus sei auf [The87] verwiesen.
Solange für die Dimension d

$$d < \log_3 \left(\frac{N-d}{2 \log_2(N-d)} \right) + 1 \tag{6.4}$$

gilt, können wir somit davon ausgehen, dass ein Fall vorliegt, welcher mit unserem Algorithmus schneller zu berechnen ist als mit einer naiven Implementierung.
Der Sortieralgorithmus benötigt während der gesamten Laufzeit maximal $O(d \cdot \log(N-d))$ Speicher und somit ist der Speicheraufwand derselbe wie beim Boxcounting Schätzer:

$$O(d \cdot (N-d)).$$

Die Approximation von d_{cor} durch

$$\hat{d}_{cor}(\epsilon_1, \epsilon_2) := \frac{\log \frac{C(\epsilon_1)}{C(\epsilon_2)}}{\log \frac{\epsilon_1}{\epsilon_2}}$$

wird analog zum Boxcounting-Schätzer (siehe Gleichung (6.2)) durchgeführt. Allerdings muss die Berechnung der Abstände nur für ϵ_{max} erfolgen. Während dieser Berechnung

[2] Da der Such-Aufwand den Aufwand der Sortierung dominiert, wird der Sortierterm vernachlässigt.

Algorithmus 2 Suche der Nachbarboxen im Korrelationsdimensionsalgorithmus

Eingabe: Nach Indizes sortierte Folge $\mathbf{X^j} = (\mathbf{x_i}, \mathcal{I}(i))_{i=j}^{N-1}, \mathbf{x_i} \in [0,1]^d$ und $\mathcal{I}(i) \in \mathbb{N}^d \ \forall i$
Ausgabe: Alle Punkte $\mathbf{x_i}$ aus Boxen, die zu $\mathbf{x_j}$ benachbart sind

```
for all ι ∈ {{−1, 0, 1}^d | ι^1 ≠ −1} do
    i ← BinarySearch(I(j) + ι)      // Suche erstes Auftreten des Index
    while I(i) = I(j) + ι do
        print x_i      // Gebe Nachbarschafts-Punkt aus
        i ← i + 1
    end while
end for
```

wird ein Histogramm

$$B_k := \frac{1}{2} \# \left\{ (\mathbf{x_i}, \mathbf{x_j}) \mid \epsilon_k \geq ||\mathbf{x_i} - \mathbf{x_j}||_2 > \epsilon_{k+1} \right\} \tag{6.5}$$

erstellt, wobei wir $\epsilon_{A+1} := 0$ gesetzt haben. Dies ermöglicht es $C(\epsilon_i)$ durch

$$C(\epsilon_i) = \sum_{k=i}^{A} B_k$$

zu errechnen. Der Rechenaufwand für das Zuweisen eines Tupels in das korrespondierende B_k ist $O(1)$, da

$$\epsilon_k \geq ||\mathbf{x_i} - \mathbf{x_j}||_2 > \epsilon_{k+1} \Leftrightarrow k+1 > \frac{\epsilon_{max}}{||\mathbf{x_i} - \mathbf{x_j}||_2 \cdot \sqrt[A]{\frac{\epsilon_{max}}{\epsilon_{min}}}} \geq k$$

gilt. Der maximale Gesamtaufwand bleibt somit unverändert.[3] Lediglich der benötigte Speicherplatz erhöht sich um $O(A)$ und beträgt somit:

$$O(A + d \cdot (N - d)).$$

Da die Laufzeit maßgeblich von $n_{max}(\epsilon_{max})$ abhängt, ist noch anzumerken, dass die Wahl eines kleinen S in (6.2) dazu führt, dass ein Gitter entsteht, in welchem eine Box sehr viele Punkte enthält. Somit besteht ein direkter Zusammenhang zwischen S und $n_{max}(\epsilon_{max})$.

[3] Anzumerken ist, dass man den Suchalgorithmus um den Faktor 3 beschleunigen kann, wenn man nur ι betrachtet, für welche $\iota^d = -1$ gilt. Aufgrund der lexikografischen Sortierung sind Fälle mit $\iota^d = 0, 1$ dann direkt zu finden ohne einen Binary-Search-Aufruf zu starten.

6.1.3 PCA

Der PCA-Algorithmus ist einfach aufgebaut. Wie in Unterabschnitt 4.2.4 beschrieben, ist es nötig die Kovarianzmatrix auszurechnen und eine Spektralzerlegung vorzunehmen. Die Einträge

$$\mathbf{C}_{i,j} = \frac{1}{N-d} \sum_{k=d}^{N-1} \mathbf{x_k}^i \mathbf{x_k}^j = \frac{1}{N-d} \sum_{k=d}^{N-1} s_{k-d+i} s_{k-d+j}$$

sind jeweils in $O(N-d)$ kalkulierbar. Da die Matrix symmetrisch ist, müssen $\frac{(d+1)\cdot d}{2}$ Einträge berechnet werden und die Laufzeit für das Aufstellen der Kovarianzmatrix ist somit

$$O(d^2 \cdot (N-d)).$$

Der Speicheraufwand für die Kovarianzmatrix beträgt $O(d^2)$.
Die Spektralzerlegung wird mittels Methoden der GSL [Gal09] realisiert. Die Laufzeit dieses Verfahrens kann von oben mit $O(d^3)$ abgeschätzt werden und der zusätzliche Speicheraufwand beträgt $O(d^2)$ für die Vektoren der Eigenbasis. Die Transformation der Vektoren in die Eigenbasis benötigt $O(d \cdot (N-d))$ Operationen und somit ist die Gesamtlaufzeit des PCA-Verfahrens abschätzbar durch

$$O(d^2 \cdot (N-d) + d^3).$$

Da in Fällen, die für uns relevant sind, stets $N > 2d$ gilt, entspricht dieser Ausdruck somit $O(d^2 \cdot (N-d))$.
Für den Fall eines solchen N beträgt der gesamte Speicheraufwand

$$O(d \cdot (N-d))$$

für das Speichern der Vektoren.
Hier zeigt sich der große Vorteil des PCA-Verfahrens: Dieses kann hohe Dimensionen und eine große Datenanzahl handhaben. Wie wir gesehen haben, ist dies für den Korrelationsdimensionsschätzer nicht der Fall. Der Boxcounting-Schätzer kann diesen Fall lediglich im Average-Case behandeln.

6.1.4 Lokale PCA

Das lokale PCA-Verfahren führt auf M verschiedenen Clustern eine gewöhnliche PCA durch. Ist m_i die Anzahl der Punkte im i-ten Cluster, so erhält man

$$O(\sum_{i=1}^{M} (d^2 \cdot m_i + d^3)) = O(d^2 \cdot (N-d) + d^3 \cdot M)$$

als Laufzeitabschätzung. Im Fall der PCA kann es vorkommen, dass es Cluster gibt, in welchen $m_i < d$ gilt. Dieser Fall ist nicht wünschenswert, da er oftmals nicht zu sinnvollen Ergebnissen führt, kann aber dennoch eintreten.
Da die Cluster sequentiell abgearbeitet werden, erhöht sich der Speicheraufwand für die Ausführung der PCA nicht. Da im Fall der lokalen PCA keine Transformation der Vektoren stattfindet, ist es nicht notwendig, die Eigenvektoren explizit zu speichern. Somit beträgt der gesamte Speicheraufwand

$$O(d \cdot (N - d)).$$

Bevor jedoch eine lokale PCA durchgeführt werden kann, ist es notwendig, die Punkte verschiedenen Clustern zuzuordnen. Wie bereits in Unterabschnitt 4.2.5 erwähnt wurde, führt das Anwenden eines guten Clustering-Algorithmus dazu, dass die resultierenden Cluster lokal, und somit räumlich begrenzt sind. Im Rahmen dieser Arbeit verwenden wir den Clustering-Algorithmus aus [BMDG05]. Die Konvergenz dieser Methode gegen ein Clustering, welches den Fehlerterm in Algorithmus 3 minimiert, wird in Proposition 3 in [BMDG05] bewiesen. Der Fehlerterm ist ein Maß für die Lokalität der Cluster.
Es sei noch angemerkt, dass der Fall $\mathbf{C_i} = \emptyset$ während der Iterationsphase des Algorithmus gesondert behandelt werden muss. Darauf haben wir in der schematischen Darstellung in Algorithmus 3 verzichtet. In unserer Implementierung werden solche Cluster verworfen.
Die Laufzeit des Clustering-Algorithmus kann durch

$$O(it \cdot (N - d) \cdot M \cdot B)$$

abgeschätzt werden, wobei it die Anzahl der Iterationen und B den Aufwand für das Auswerten der Bregman-Divergenz $d_b(\cdot, \cdot)$ bezeichnet. Ist $d_b(\cdot, \cdot)$ der euklidische Abstand, so ist $B = d$.
Der Clustering-Algorithmus benötigt $O(M \cdot d)$ zusätzlichen Speicherplatz, um die Clustermittelpunkte zu speichern. Somit erhalten wir

$$O(\underbrace{d^2 \cdot (N - d) + d^3 \cdot M}_{\text{PCA auf den Clustern}} + \underbrace{it \cdot (N - d) \cdot M \cdot B}_{\text{Clustering}})$$

als Laufzeitabschätzung und

$$O(d \cdot (N - d) + M \cdot d) = O(d \cdot ((N - d) + M)) = O(d \cdot (N - d))$$

als Speicherbedarf für die lokale PCA.[4]

[4] Dies ist darauf zurückzuführen, dass $M > (N - d)$ keine sinnvollen Ergebnisse erzielen kann. Somit schließen wir diesen Fall aus.

Algorithmus 3 Clustering mit Bregman-Divergenzen

Eingabe: Einen Abbruchparameter $\epsilon \in \mathbb{R}$, eine Menge $\mathbf{X} = \{\mathbf{x_i} \in [0,1]^d\}_{i=d}^{N-1}$, eine Bregman-Divergenz $d_b : [0,1]^2 \to \mathbb{R}^+$, die gewünschte Clusteranzahl M

Ausgabe: M disjunkte Clustermengen $\mathbf{C_i}$, sodass $\cup_{i=1}^{M} \mathbf{C_i} = \mathbf{X}$

for $i = 1, \ldots, M$ **do**
  $\mathbf{c_i} \leftarrow \text{random}(\mathbf{X})$ // Initialisiere $\mathbf{c_i}$ mit einem zufälligen Punkt aus $\mathbf{X}$
end for
error $\leftarrow \infty$
olderror $\leftarrow 1$
while $\frac{\text{olderror}-\text{error}}{\text{olderror}} > \epsilon$ **do**
  olderror $\leftarrow$ error
  for $i = 1, \ldots, M$ **do**
    $\mathbf{C_i} \leftarrow \{\mathbf{x_j} \in \mathbf{X} \mid i = \min\{\arg\min_i d_b(\mathbf{x_j}, \mathbf{c_i})\}\}$
    // $\mathbf{C_i}$ ist Menge aller Punkte $\mathbf{x_j}$ mit $d_b(\mathbf{x_j}, \mathbf{c_i}) < d_b(\mathbf{x_j}, \mathbf{c_k})\ \forall k < i$
    // bzw. $d_b(\mathbf{x_j}, \mathbf{c_i}) \leq d_b(\mathbf{x_j}, \mathbf{c_k})\ \forall k > i$
  end for
  for $i = 1, \ldots, M$ **do**
    $\mathbf{c_i} \leftarrow \sum_{\mathbf{x} \in \mathbf{C_i}} \frac{\mathbf{x}}{|\mathbf{C_i}|}$ // Setze $\mathbf{c_i}$ auf das Mittel aller Elemente aus $\mathbf{C_i}$
  end for
  error $\leftarrow \sum_{i=1}^{M} \sum_{\mathbf{x} \in \mathbf{C_i}} d_b(\mathbf{x}, \mathbf{c_i})$
    // Abweichungen von den Clustermittelpunkten sind Fehlermaß
end while
return $(\mathbf{C_i})_{i=1}^{M}$

6.2 Delay-Schätzer

6.2.1 Autokorrelation

Um die Approximation

$$\hat{C}(\tau) := \frac{1}{\hat{\sigma}^2} \cdot \frac{1}{N-\tau-1} \sum_{i=\tau+1}^{N} \left(\left(\nu_i - \frac{1}{N} \sum_{j=1}^{N} \nu_j \right) \left(\nu_{i-\tau} - \frac{1}{N} \sum_{j=1}^{N} \nu_j \right) \right)$$

aus (4.35) auszurechnen, benötigt man $O(N)$ Operationen. Wird $C(\tau)$ für $\tau = 1, \ldots, T$ ausgerechnet, so ist die gesamte Laufzeit durch

$$O(N \cdot T)$$

begrenzt. Der Speicheraufwand beträgt $O(N)$, da lediglich auf die Zeitreihendaten zugegriffen werden muss.

6.2.2 Mutual Information

Das Aufstellen der Histogramme

$$\begin{aligned} \hat{\mu}(I_k) &= \tfrac{1}{N}\#\{n \in \{1,\dots,N\} \mid \nu_n \in I_k\} \\ \hat{\gamma}_\tau(I_i, I_j) &= \tfrac{1}{N-\tau}\#\{n \in \{\tau+1,\dots,N\} \mid \nu_n \in I_i \text{ und } \nu_{n-\tau} \in I_j\} \end{aligned}$$

aus (4.40) und das anschließende Auswerten nach Gleichung (4.38) wird in unserem Algorithmus in Laufzeit $O(N + K^2)$ realisiert, wobei $K = \frac{1}{\epsilon}$ die Anzahl der Partitionen I_k ist, in die das Intervall unterteilt wird. Errechnet man die Mutual Information $M_\epsilon(\tau)$ für $\tau = 1, \dots, T$, so erhält man eine Gesamtlaufzeit von

$$O(N + T \cdot K^2),$$

da die Zuordnung der Punkte zum korrespondierendem Intervall in einem Vorverarbeitungsschritt geschehen kann. Der gesamte Speicheraufwand beträgt $O(N + K^2)$.

6.3 Vorhersage mit dünnen Gittern

Statt $\mathbf{x_i}$ bezeichne im Folgenden $\mathbf{x}_i$ den i-ten Datenpunkt, um Verwechslungen zwischen Multiindizes und Indizes zu vermeiden.

6.3.1 Reguläre dünne Gitter

Der verwendete Algorithmus stammt aus [Gar04]. Wir erläutern nun kurz die Vorgehensweise und die Komplexität.
Das reguläre dünne Gitter Ω_t^s besitzt $O(2^t \cdot t^{d-1})$ Freiheitsgrade, wobei die Konstante allerdings von d abhängt. Durch den Ansatz mittels der Kombinationstechnik müssen nun $O(dt^{d-1})$ Probleme der Ordnung $O(2^t)$ gelöst werden. Für jedes involvierte Gitter $\Omega_\mathbf{l}$ sind nun die Koeffizienten $\hat{\alpha}_{\mathbf{l},\mathbf{i}}$ aus der Darstellung

$$\hat{f}_\mathbf{l} = \sum_{\mathbf{i} \le 2^\mathbf{l}} \hat{\alpha}_{\mathbf{l},\mathbf{i}} \phi_{\mathbf{l},\mathbf{i}}$$

zu berechnen. Dies geschieht im Gleichungssystem

$$\left(\mathbf{B_l}^T \mathbf{B_l} + (N-d)\lambda \mathbf{C_l}\right) \hat{\alpha}_\mathbf{l} = \mathbf{B_l}^T \mathbf{s}, \tag{6.6}$$

siehe auch (5.14). Bezeichnet $G_\mathbf{l} = \prod_{j=1}^d \left(2^{l_j} + 1\right)$ die Anzahl der Gitterpunkte, so ist $(\hat{\alpha}_\mathbf{l})_{\mathbf{j}=\mathbf{0}}^{2^\mathbf{l}} \in \mathbb{R}^{G_\mathbf{l}}$ der Vektor der Unbekannten und $\mathbf{s} = (s_{d+1}, \dots, s_N)^T \in \mathbb{R}^{N-d}$ der Zeitreihenvektor.

$\mathbf{B_l} \in \mathbb{R}^{(N-d)\times G_\mathbf{l}}$ stellt die Datenmatrix

$$(\mathbf{B_l})_{i,\mathbf{j}} = \phi_{\mathbf{l},\mathbf{j}}(\mathbf{x}_i)$$

dar und $\mathbf{C_l} \in \mathbb{R}^{G_\mathbf{l}\times G_\mathbf{l}}$ ist die Regularisierungsmatrix

$$(\mathbf{C_l})_{\mathbf{i},\mathbf{j}} = \sum_{|\mathbf{a}|_*=1} \left(D^\mathbf{a}\phi_{\mathbf{l},\mathbf{i}}, D^\mathbf{a}\phi_{\mathbf{l},\mathbf{j}}\right)_{L_2},$$

mit

- $* = l_1$ für die H^1- und
- $* = l_\infty$ für die H^1_{mix}-Regularisierung.

Da jeder Datenpunkt $\mathbf{x}_i$ im Träger von maximal 2^d Basisfunktionen liegt und die Auswertung einer solchen in $O(d)$ durchführbar ist, benötigt man zum Aufstellen sämtlicher Einträge

$$\left(\mathbf{B_l}^T\mathbf{B_l}\right)_{\mathbf{j},\mathbf{k}} = \sum_{i=d}^{N-1} \phi_{\mathbf{l},\mathbf{j}}(\mathbf{x}_i)\cdot\phi_{\mathbf{l},\mathbf{k}}(\mathbf{x}_i)$$

$O(2^d \cdot 2^d \cdot d \cdot (N-d)) = O(d \cdot 2^{2d}(N-d))$ Operationen. Wie in (5.16) gesehen, beträgt die maximale Anzahl an Basisfunktionen, die nichtleeren Schnitt mit einer festen Basisfunktion $\phi_{\mathbf{l},\mathbf{i}}$ in der nodalen Basis haben $O(3^d)$ und der gesamte Speicheraufwand für die Matrix $\mathbf{B_l}^T\mathbf{B_l}$ betrgt somit $O(3^d \cdot G_\mathbf{l})$, falls nur Einträge gespeichert werden, die nicht Null sind. Analog ist das Aufstellen der rechten Seite in $O(d \cdot 2^d(N-d))$ realisierbar, wobei der resultierende Vektor lediglich $O(N-d)$ Speicherplatz benötigt.
Nach diesen Überlegungen ist es offensichtlich, dass das Aufstellen der Regularisierungsmatrix $\mathbf{C_l}$ mit einer Komplexität von $O(c(*)\cdot d \cdot 3^d \cdot G_\mathbf{l})$ durchführbar ist und einen Speicherplatz von $O(3^d \cdot G_\mathbf{l})$ benötigt. $c(*)$ ist hierbei die Anzahl der Summanden des Regularisierungsoperators. Für die H^1-Seminorm ist $c(*) = d$, da in jede Koordinatenrichtung einmal differenziert wird. Die H^1_{mix}-Seminorm hingegen führt zu einem größeren Aufwand, da die Anzahl der Summanden hier

$$c(*) = \#\{\mathbf{a} \in \mathbb{N}^d \mid |\mathbf{a}|_\infty = 1\} = 2^d - 1$$

ist.
Das Lösen des Gleichungssystems wird mittels eines diagonal-vorkonditionierten CG-Verfahrens realisiert und benötigt somit maximal $O(G_\mathbf{l})$ Iterationen, in denen Matrix-Vektor-Multiplikationen mit $O(3^d \cdot G_\mathbf{l})$ Aufwand durchgeführt werden müssen, da dies der Anzahl der Nicht-Null-Elemente der Matrizen entspricht. [5]

[5] In [Gar04] wird darauf hingewiesen, dass das Lösen mittels eines CG-Verfahren zwar nicht optimal ist, ein für diese Zwecke geeignetes hochdimensionales Mehrgitterverfahren allerdings noch nicht zur Verfügung steht. Das Lösen des Gleichungssystems soll in dieser Arbeit nicht thematisiert werden.

Der gesamte Rechenaufwand ist somit von oben durch

$$\underbrace{O(d \cdot 2^{2d}(N-d) + c(*) \cdot d \cdot 3^d \cdot G_{\mathbf{l}})}_{\text{Aufstellen der Matrizen}} + \underbrace{O(3^d \cdot G_{\mathbf{l}}^2)}_{\text{CG-Algorithmus}}$$

abschätzbar, wobei die Anzahl der Iterationen des CG-Algorithmus bei guter Vorkonditionierung allerdings $O(1)$ ist. Der gesamte Speicheraufwand beträgt

$$O(3^d \cdot G_{\mathbf{l}}).$$

Die erhaltenen Funktionen werden nun mittels der Kombinationstechnik

$$f_t^c(\mathbf{x}) := \sum_{q=0}^{d-1} (-1)^q \begin{pmatrix} d-1 \\ q \end{pmatrix} \sum_{|\mathbf{l}|_1 = n-q} \hat{f}_{\mathbf{l}}(\mathbf{x}) \tag{6.7}$$

aus (5.21) aufaddiert und man erhält somit die Möglichkeit, die rekonstruierte Funktion f_t^c auf neuen Datenpunkten auszuwerten. Die Gesamtkomplexität ergibt sich nun als Summe der einzelnen Komplexitäten.
Die Auswertung von M Datenpunkten $(\tilde{\mathbf{x}}_i)_{i=1}^M$ ist mit einem Rechenaufwand von $O(M \cdot d \cdot t^{d-1} \cdot 2^d)$ realisierbar. Der Faktor 2^d stammt hierbei von den Basisfunktionen $\phi_{\mathbf{l},\mathbf{i}}$, in deren Träger der Punkt $\tilde{\mathbf{x}}_j$ liegt. Die gesamte Vorgehensweise ist in Algorithmus 4 abgebildet.
In [Gar04] wird die Möglichkeit erwähnt, die Matrizen nicht direkt aufzustellen, sondern ein *on-the-fly*-Aufstellen zu realisieren, welches mit dem CG-Verfahren vereinbar ist. Dies ist in großen Dimensionen hilfreich, wenn die Matrizen zu groß für den Arbeitsspeicher sind. Dieser Ansatz ist allerdings sehr rechenintensiv und sollte nur genutzt werden, falls die Matrizen nicht mehr in den Arbeitsspeicher eingelesen werden können. Des Weiteren wird erwähnt, dass die Matrix $\mathbf{B}_{\mathbf{l}}$ direkt aufgestellt werden kann, ohne dass ${\mathbf{B}_{\mathbf{l}}}^T \mathbf{B}_{\mathbf{l}}$ aufgestellt werden muss. Dieser Ansatz ist für eine on-the-fly-Berechnung günstiger. Allerdings sind diese Alternativen nur für eine moderate Datenanzahl $N - d$ sinnvoll. Für detailliertere Betrachtungen sei auf [Gar04] verwiesen.

6.3.2 Dimensionsadaptive dünne Gitter

Die Laufzeit des dimensionsadaptiven Algorithmus hängt maßgeblich von der konstruierten Indexmenge $\mathbf{I}$ und dem Aufwand des Fehlerindikators ab. Eine genaue Analyse der Laufzeit ist fallabhängig, jedoch kann die Laufzeit der Auswertung der Funktion auf den M Testdaten durch $O(M \cdot L \cdot 2^d)$ angegeben werden, wobei

$$L = \# \left\{ \mathbf{l} \in \mathbf{I} \mid \sum_{\mathbf{k}=\mathbf{0}}^{\mathbf{1}} (-1)^{|\mathbf{k}|_1} \chi^{\mathbf{I}}(\mathbf{l}+\mathbf{k}) \neq 0 \right\}$$

Algorithmus 4 Approximieren von g mit der Kombinationstechnik – siehe auch [Gar04]

Eingabe: Die eingebetetten Trainingsdaten $(\mathbf{x}_i)_{i=d}^{N-1} \in \mathbb{R}^{(N-d)\times d}$, die Zeitreihenpunkte $(s_j)_{j=d+1}^{N} \in \mathbb{R}^{N-d}$, das multivariate Verfeinerungslevel $\mathbf{l} \in \mathbb{N}^d$, Testdaten $(\tilde{\mathbf{x}}_i)_{i=1}^{M}$
Ausgabe: Die Werte der rekonstruierten Funktion $(f_{\mathbf{I}}^c(\tilde{\mathbf{x}}_i))_{i=1}^{M} \in \mathbb{R}^M$ auf den Testdaten

```
    // Errechnen der Approximation:
for all l ∈ I do
    if (Σ_{k=0}^{1} (−1)^{|k|_1} χ^I(l + k)) ≠ 0 then
        Löse (B_l^T B_l + (N − d)λC_l) α̂_l = B_l^T s mit CG-Verfahren
        Speichere α̂_l
    end if
end for
    // Auswerten mittels Kombinationstechnik:
for i = 1, ..., M do
    ŷ_i ← 0
    for all l ∈ I do
        if (Σ_{k=0}^{1} (−1)^{|k|_1} χ^I(l + k)) ≠ 0 then
            ŷ_i ← ŷ_i + Σ_{k=0}^{1} (−1)^{|k|_1} χ^I(l + k) · f̂_l(x̃_i)
        end if
    end for
end for
return (ŷ_i)_{i=1}^{M}
```

gilt. Die Vorschrift zur Bestimmung der zu verwendenden Indexmenge $\mathbf{I}$ findet sich in Algorithmus 5 und ist analog zu [GG03]. Eine Illustration der Vorgehensweise ist in Abbildung 6.1 zu sehen.
Wir wollen nun Fehlerindikatoren $\epsilon_{\mathbf{k}}$ für ein multivariates Level $\mathbf{k}$ vorstellen. Der einfachste Indikator hängt direkt mit der Darstellung der rekonstruierten Funktion $f_{\mathbf{I}\cup\{\mathbf{k}\}}^c$ zusammen, wobei $\mathbf{I}$ die Menge der bereits verwendeten Indizes darstellt, siehe auch Algorithmus 5. Stellen wir die rekonstruierte Funktion in der hierarchischen Basis durch

$$f_{\mathbf{I}\cup\{\mathbf{k}\}}^c = \sum_{\phi_{\mathbf{l},\mathbf{i}} \in V_{\mathbf{I}\cup\{\mathbf{k}\}}} \alpha_{\mathbf{l},\mathbf{i}} \phi_{\mathbf{l},\mathbf{i}}$$

dar, so können wir durch

$$\epsilon_{\mathbf{k}}^{\text{hier}} := \frac{1}{\#\{\phi_{\mathbf{l},\mathbf{i}} \mid \phi_{\mathbf{l},\mathbf{i}} \in V_{\mathbf{I}\cup\{\mathbf{k}\}} \setminus V_{\mathbf{I}}\}} \sum_{\phi_{\mathbf{l},\mathbf{i}} \in V_{\mathbf{I}\cup\{\mathbf{k}\}} \setminus V_{\mathbf{I}}} |\alpha_{\mathbf{l},\mathbf{i}}| \cdot ||\phi_{\mathbf{l},\mathbf{i}}||_*$$

einen Fehlerindikator definieren. Dieser beruht auf der Beobachtung, dass die hierarchischen Koeffizienten mit den zweiten Ableitungen der zu approximierenden Funktion

Algorithmus 5 Dimensionsadaptive Bestimmung der Indexmenge $\mathbf{I}$ – siehe auch [Gar04]

Eingabe: Die eingebetetten Trainingsdaten $(\mathbf{x}_i)_{i=d}^{N-1} \in \mathbb{R}^{(N-d)\times d}$, die Zeitreihenpunkte $(s_j)_{j=d+1}^{N} \in \mathbb{R}^{N-d}$, lokale Fehlerindikatoren $\epsilon_{\mathbf{k}}$, ein globaler Fehlerindikator E, eine globale Toleranzgrenze tol $\in \mathbb{R}^+$

Ausgabe: Eine zulässige Indexmenge $\mathbf{I} \subset \mathbb{N}^d$

$\mathbf{I} \leftarrow \emptyset$
$\mathcal{A} \leftarrow \{\mathbf{0}\}$ // Menge der *aktiven* Indizes
while tol $< E$ **do**
  $\mathbf{i} \leftarrow \arg\max_{\mathbf{j}\in\mathcal{A}} \epsilon_{\mathbf{j}}$ // Wähle Index mit maximalem Beitrag
  $I \leftarrow I \cup \{\mathbf{i}\}$
  $\mathcal{A} \leftarrow \mathcal{A} \setminus \{\mathbf{i}\}$
  for $k = 1, \ldots, d$ **do**
    $\mathbf{j} \leftarrow \mathbf{i} + \mathbf{e}_k$ // Suche in jeder Richtung
    // Überprüfe Zulässigkeitsbedingung:
    if $\mathbf{j} - \mathbf{e}_l \in \mathbf{I}$ für alle $l = 1, \ldots, d$ **then**
      $\mathcal{A} \leftarrow \mathcal{A} \cup \{\mathbf{j}\}$
      Berechne $\hat{f}_{\mathbf{j}}$
    end if
  end for
  for all $\mathbf{k} \in \mathbf{I} \cup \mathcal{A}$ **do**
    Aktualisiere $\epsilon_{\mathbf{k}}$ // Laufzeit hängt von den Indikatoren ab
  end for
  Aktualisiere E // Laufzeit hängt vom Indikator ab
end while
return $\mathbf{I}$

korrelieren und somit ein Maß für die Glattheit der Funktion darstellen, siehe [BG04]. Die Norm $||\cdot||_*$, in welcher die Basisfunktionen gemessen werden, sollte als die Norm gewählt werden, in welcher man eine möglichst gute Approximation wünscht. Es sei angemerkt, dass zur Auswertung des Fehlerindikators die entsprechenden Funktionen in die hierarchische Basis transformiert werden müssen. Wie in [Gar04] erwähnt, ist dieser Fehlerindikator allerdings ungeeignet, falls man eine Hierarchie mit konstanten Funktion zugrundelegt.
Ein alternativer Fehlerindikator kann dadurch konstruiert werden, dass die l_2-Datenfehler aus dem Funktional (5.13) verglichen werden:

$$\epsilon_{\mathbf{k}}^{l_2} := \left| \frac{1}{N-d} \sum_{i=d}^{N-1} (s_{i+1} - f^c_{\mathbf{I}\cup\{\mathbf{k}\}}(\mathbf{x_i}))^2 \right| - \left| \frac{1}{N-d} \sum_{i=d}^{N-1} (s_{i+1} - f^c_{\mathbf{I}}(\mathbf{x_i}))^2 \right|$$

Da dieser Ausdruck negativ sein kann, stellt der Indikator $|\epsilon_{\mathbf{k}}^{l_2}|$ eine sinnvolle Alternative

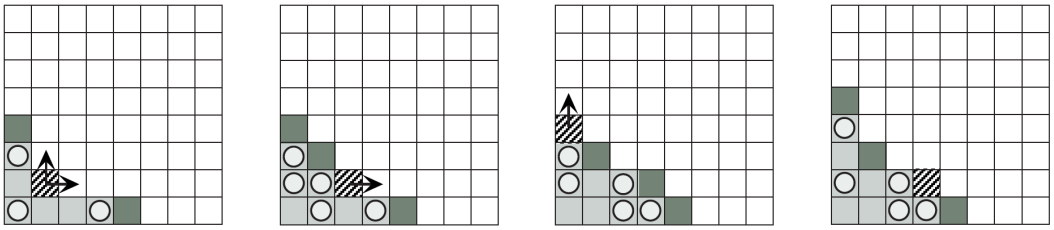


Abb. 6.1: Vier Iterationsschritte des dimensionsadaptiven Dünngitteralgorithmus für $d = 2$. Die Indizes in $\mathbf{I}$ sind hellgrau unterlegt, die Indizes in $\mathcal{A}$ dunkelgrau. Der Index $\mathbf{i}$ aus $\mathcal{A}$ für welchen der Fehlerindikator am größten ist, ist gestreift dargestellt. Die Pfeile geben die Indizes an, welche während des Iterationsschrittes zu $\mathcal{A}$ hinzugefügt werden – hierzu wird die Zulässigkeitsbedingung überprüft, siehe Algorithmus 5. Die Indizes, welche einen Nicht-Null-Beitrag zur Kombinationsformel liefern, sind mit einem Kreis markiert. Abbildung entnommen aus [Gar04].

dar. Ein weiterer Fehlerindikator wird durch die Änderung der rekonstruierten Funktion auf den Datenpunkten definiert:

$$\epsilon_{\mathbf{k}}^{f} := \left| \frac{1}{N-d} \sum_{i=d}^{N-1} (f_{\mathbf{I}}^{c}(\mathbf{x_i}) - f_{\mathbf{I}\cup\{\mathbf{k}\}}^{c}(\mathbf{x_i}))^2 \right|$$

Den globalen Indikator wählen wir abhängig von den verwendeten lokalen Fehlerindikatoren als

$$E := \sum_{\mathbf{l}\in\mathcal{A}} \epsilon_{\mathbf{l}},$$

wobei $\mathcal{A}$ die Menge der aktiven Indizes ist, siehe Algorithmus 5.
Neben der im vorherigen Unterabschnitt beschriebenen höheren Laufzeit einer H^1_{mix}-Regularisierung im Vergleich zur H^1-Variante, sei erwähnt, dass der Raum, in dem die Lösung erzielt werden kann, bei einem RKHS-regularisierten Verfahren bereits festgelegt ist. Im dimensionsadaptiven Algorithmus wäre es allerdings sinnvoll, den Lösungsraum selbst anpassen zu können, da dieser z.B. die Struktur eines gewichteten Anker-Sobolevraums wie in (5.24) haben kann. In [Gar04] wird argumentiert, dass die Wahl des einfachereren H^1-Operators in diesem Zusammenhang sinnvoll sein kann.

6.3.3 Ortsadaptive dünne Gitter

Der ortsadaptive Algorithmus entstand durch Modifikation des Codes aus [Feu10, FG09]. Um die Struktur der ortsadaptiven dünnen Gitter effizient zu speichern wurden spezielle *Hash*-Tabellen verwendet, welche es ermöglichen, ein Gitter mit K Punkten mit einem

Speicherbedarf von $O(K)$ so zu verwalten, dass das Zugreifen auf einen beliebigen Punkt in $O(d)$ zu bewerkstelligen ist. Wir wollen hier nicht näher auf diese Konzepte eingehen und verweisen auf [Gri98, Feu10].
Die Vorgehensweise zur Bestimmung der Indexmenge ist in Algorithmus 6 zu sehen.

Algorithmus 6 Ortsadaptive Bestimmung des Gitters Ω^{adp}

Eingabe: Die eingebetetten Trainingsdaten $(\mathbf{x}_i)_{i=d}^{N-1} \in \mathbb{R}^{(N-d)\times d}$, die Zeitreihenpunkte $(s_j)_{j=d+1}^{N} \in \mathbb{R}^{N-d}$, lokale Fehlerindikatoren $\epsilon_{\mathbf{x}}$, eine Toleranzgrenze tol $\in \mathbb{R}^+$, ein maximales Level $L \in \mathbb{N}$, ein Anfangslevel $l \in \mathbb{N}$
Ausgabe: Ein ortsadaptives Gitter $\Omega^{\text{adp}} \subset [0,1]^d$

$\Omega^{\text{adp}} \leftarrow \Omega_l^s$
for $k = l, \ldots, L$ **do**
  for all $\mathbf{x} \in \Omega^{\text{adp}}$ **do**
    if $\epsilon_{\mathbf{x}} >$ tol **then**
      for $j = 1, \ldots, d$ **do**
        $\Omega^{\text{adp}} \leftarrow \Omega^{\text{adp}} \cup \left\{\mathbf{x} \pm 2^{-k} \cdot \mathbf{e}_j\right\}$
        Füge fehlende hierarchische Vorfahren ein
      end for
    end if
  end for
  for all $\mathbf{x} \in \Omega^{\text{adp}}$ **do**
    Aktualisiere $\epsilon_{\mathbf{x}} \forall\ \mathbf{x} \in \Omega^{\text{adp}}$ // Laufzeit hängt von den Indikatoren ab
  end for
end for
return Ω^{adp}

Die Fehlerindikatoren sind nun punktweise zu wählen. Analog zum dimensionsadaptiven Fehlerindikator, ist auch hier ein Indikator konstruierbar, der die hierarchischen Koeffizienten berücksichtigt:

$$\epsilon_{\mathbf{x}_{\mathbf{l},\mathbf{i}}}^{\text{hier}} := |\alpha_{\mathbf{l},\mathbf{i}}| \cdot ||\phi_{\mathbf{l},\mathbf{i}}||_*.$$

Wir weisen hierbei einem Punkt $\mathbf{x} \in \Omega^{\text{adp}}$ das korrespondierende Level und den korrespondierenden Index in der hierarchischen Darstellung zu. Die l_2-Fehlerindikatoren aus dem dimensionsadaptiven Algorithmus sind hier nicht mehr sinnvoll.

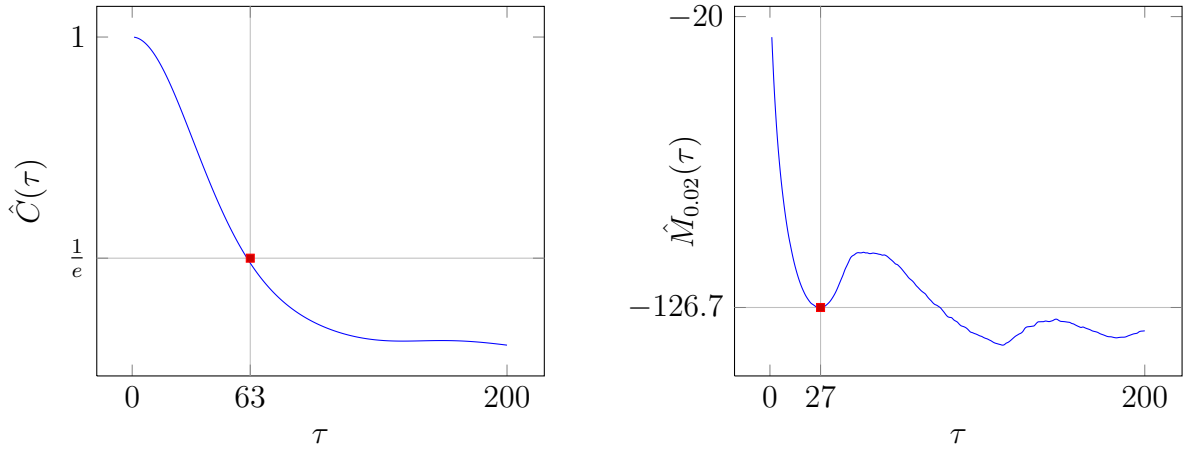
7 Experimente

In diesem Kapitel werden die kennengelernten Konzepte sowohl auf synthetischen als auch anwendungsbezogenen Datensätzen getestet. Zunächst werden die in dieser Arbeit vorgestellten Algorithmen auf das bekannte Lorenz-System angewendet. Damit unsere Algorithmen ausgeführt werden können, wurden die Zeitreihen mit Hilfe einer linearen Transformation auf $[0, 1]$ skaliert. Die s_i im Lernfunktional wurden allerdings nicht skaliert.

7.1 Konvergenz der Verfahren

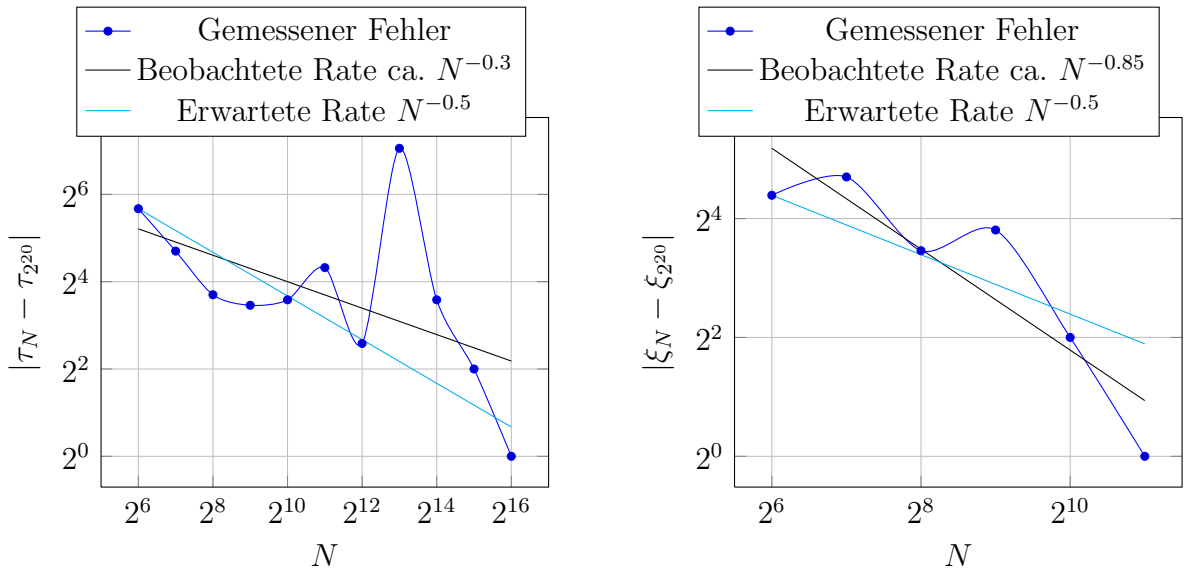
7.1.1 Delay-Schätzer am Beispiel des Lorenz-Systems

Unser erstes Beispiel soll nicht nur das Konvergenzverhalten der Delay-Schätzer veranschaulichen, sondern auch verdeutlichen, dass es sinnvoll ist, diese zu verwenden, um vor dem Einbetten der Zeitreihe die vorliegenden Daten auf Korrelationen zu überprüfen. Zu diesem Zweck wenden wir uns dem bereits bekannten Lorenz-System zu.
Die vorliegenden Zeitreihen T_N bestehen aus $N = 2^i, i = 6, \ldots, 20$ Werten und wurden durch ein Runge-Kutta-Verfahren vierter Ordnung mittels $o((x_1, x_2, x_3)^T) = x_1$ aus dem Lorenz-System (2.1) gewonnen ($a = 10$, $b = 28$, $c = \frac{8}{3}$, $\mathbf{z_0} = (1, 1, 1)^T$). Entgegen der bisher betrachteten Beispiele des Lorenz-Systems wurde beim Runge-Kutta-Verfahren eine Schrittweite von 0.005 – statt 0.1 – verwendet. Dies führt zu stärkeren Korrelationen zwischen konsekutiven, durch die Observable gewonnenen Zeitreihenwerten.
Für jede der Zeitreihen wird $\tau_N = \min\{t \in \mathbb{N} \mid \hat{C}(t) < \frac{1}{e}\}$ als Resultat der Autokorrelationsmethode gesucht. In Abbildung 7.1 (a) ist die Autokorrelationsfunktion für $N = 2^{20}$ zu sehen. Die Analyse ergibt, dass $\tau_{2^{20}} = 63$ ist. Die Werte $|\tau_N - \tau_{2^{20}}|$ sind im Konvergenzplot 7.2 (a) zu sehen. Die beobachtete Rate wurde hierbei durch einen linearen *Least-Squares-Fit* errechnet. Es wurden nur Punkte mit echt positivem Fehler in Betracht gezogen.
Eine Unterteilung des Intervalls $[0, 1]$ in 50 disjunkte Intervalle $[0, 1] = \cup_{i=1}^{50} I_k$ und eine anschließende Bestimmung der ersten Minimalstelle ξ_N der Mutual Information-Funktion $\hat{M}_{0.02}$ führt für $N = 2^{20}$ zu $\xi_{2^{20}} = 27$, wie in Abbildung 7.1 (b) zu sehen ist. Der Konvergenzplot ist in Abbildung 7.2 (b) zu finden.
Für den Autokorrelationsschätzer gilt $\tau_{17} = \cdots = \tau_{20} = 63$. Der Mutual Information-Schätzer konvergiert früher und es gilt $\xi_{12} = \cdots = \xi_{20} = 27$.
Bei Betrachtung des Konvergenzplots der Autokorrelationsfunktion fällt sofort auf, dass



(a) Autokorrelation (b) Mutual Information

Abb. 7.1: Autokorrelation und Mutual Information für das Lorenz-System mit Runge-Kutta-Schrittweite 0.005



(a) Autokorrelationsschätzer (b) Mutual Information-Schätzer

Abb. 7.2: Konvergenzverhalten der Delay-Schätzer

der Fehler für $N = 2^{13}$ wesentlich größer ist als für $N = 2^{12}$. Dies erscheint unintuitiv, da eine Verdopplung von N hier zu einem größeren Fehler führt. Eine Erklärung für dieses Phänomen findet sich im Verhalten der Zeitreihe. Wie in Abbildung 7.4 zu sehen ist, liegt der Erwartungswert von $T_{2^{12}}$ ungefähr bei 0.3 und die Werte $T_{2^{12}}(t) - 0.3$ und $T_{2^{12}}(t - \tau) - 0.3$ haben – auch für kleine τ – für viele t verschiedene Vorzeichen. Der

2^i	$\lvert\tau_{2^i} - \tau_{2^20}\rvert$	$\frac{\lvert\tau_{2^{i+1}} - \tau_{2^20}\rvert}{\lvert\tau_{2^i} - \tau_{2^20}\rvert}$	$\lvert\xi_{2^i} - \xi_{2^20}\rvert$	$\frac{\lvert\xi_{2^{i+1}} - \xi_{2^20}\rvert}{\lvert\xi_{2^i} - \xi_{2^20}\rvert}$
64	51	1.96	21	0.81
128	26	2.00	26	2.36
256	13	1.18	11	0.79
512	11	0.92	14	3.50
1,024	12	0.60	4	4.00
2,048	20	3.33	1	–
4,096	6	0.05	0	–
8,192	133	11.08	0	–
16,384	12	3.00	0	–
32,768	4	4.00	0	–
65,536	1	–	0	–
131,072	0	–	0	–
262,144	0	–	0	–
524,288	0	–	0	–
1,048,576	0	–	0	–

Tab. 7.3: Fehler der Delay-Schätzer

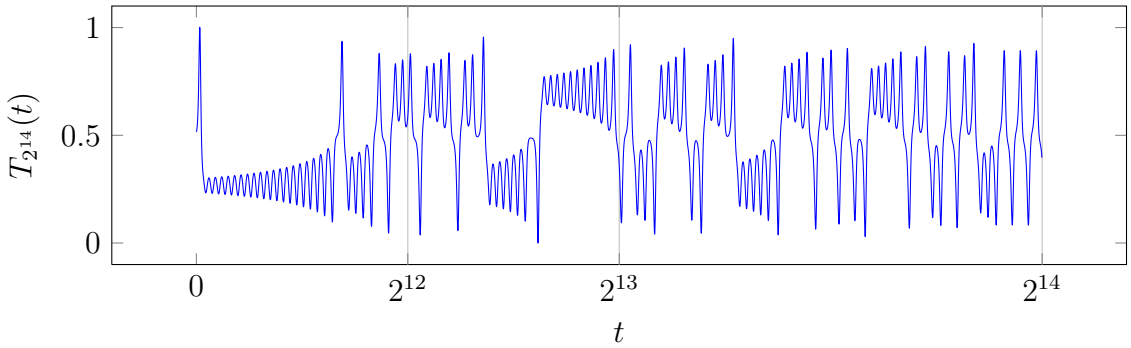
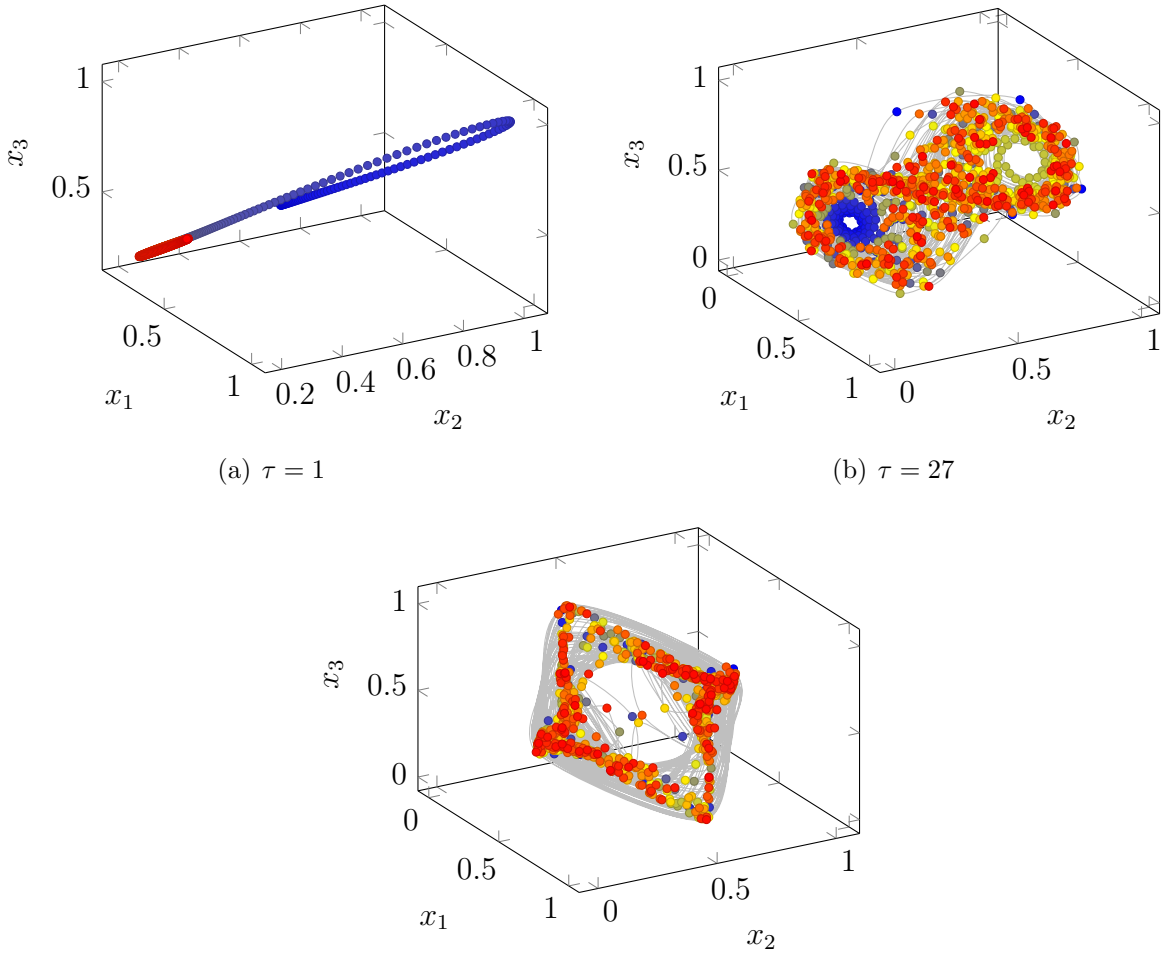


Abb. 7.4: Die Zeitreihe $T_{2^{14}}$

Erwartungswert der Zeitreihe $T_{2^{13}}$ hingegen liegt bei ca. 0.45 und für kleine τ haben $T_{2^{13}}(t) - 0.45$ und $T_{2^{13}}(t-\tau) - 0.45$ für viele t dasselbe Vorzeichen. Dies führt zu einem großen Wert $C(\tau)$. Der Erwartungswert von $T_{2^{14}}$ ist ebenfalls ungefähr 0.45, allerdings wechselt $T_{2^{14}}(t) - 0.45$ für $t > 2^{13}$ wesentlich häufiger das Vorzeichen, was zu kleineren Werten von $C(\tau)$ führt.
Offenbar ist dies eine Eigenart der Zeitreihe, die der Autokorrelationsschätzer nicht handhaben kann. Die Mutual Information ist allerdings in der Lage auch in dieser Situation eine stabile Schätzung zu liefern.
Die Konvergenzrate des Mutual Information-Schätzers ist in diesem Beispiel ebenfalls



(a) $\tau = 1$

(b) $\tau = 27$

(c) $\tau = 63$

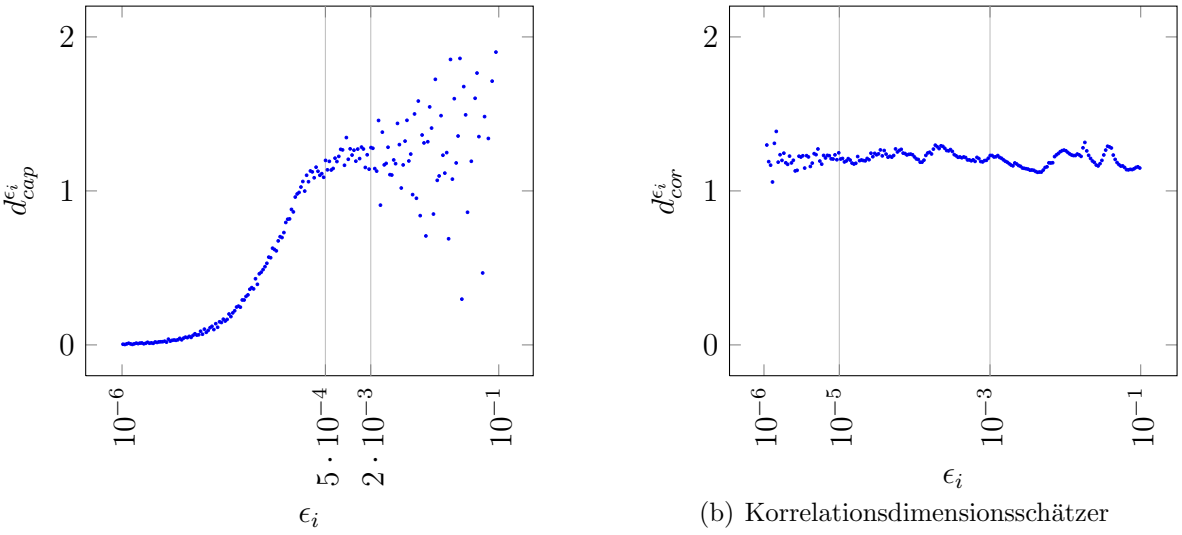
Abb. 7.5: Rekonstruktionen des Lorenz-Attraktors aus der Zeitreihe $T_{2^{20}}$ mit verschiedenen Delay-Parametern – Der zeitliche Verlauf ist durch den Farbverlauf von blau nach rot gekennzeichnet. Es werden jeweils die ersten 1000 Punkte der Trajektorie gezeigt.

deutlich höher als beim Autokorrelationsschätzer.
Es fällt außerdem auf, dass die beiden Verfahren sehr unterschiedliche Schätzungen für den optimalen Delay-Parameter liefern. In Abbildung 7.5 sind Einbettungen des rekonstruierten Attraktors für verschiedene Delay-Parameter zu sehen.
Wie zu sehen ist, ist eine Rekonstruktion mit $\tau = 1$ nicht sinnvoll, da die Punkte nahe der Raumdiagonalen $\{x_1 = x_2 = x_3\}$ liegen und die ursprüngliche Struktur des Attraktors nicht mehr zu erkennen ist.[1] Die Rekonstruktion für $\tau = 27$ hingegen erfasst die Attraktor-Struktur und lässt Rückschlüsse auf die Trajektorie des ursprünglichen Prozesses zu. Die Rekonstruktion, die wir für $\tau = 81$ erhalten, scheint nicht mehr mit dem ursprünglichen Prozess zusammenzuhängen und die Struktur des Lorenz-Attraktors lässt sich nicht erkennen.
Die Mutual Information ist in diesem Beispiel in der Lage, ein τ zu liefern, welches eine gute Rekonstruktion ermöglicht. Das Abfallen der Autokorrelationsfunktion auf $\frac{1}{e}$ hingegen, scheint keinen Hinweis auf einen adäquaten Zeitparameter τ zu liefern. Dies zeigt, dass es nötig ist, die Ergebnisse der Delay-Schätzer visuell zu überprüfen. Erhalten wir ein Ergebnis, welches um die Raumdiagonale konzentriert ist, so scheint es, dass ein größerer Wert von τ eine bessere Alternative ist. Sind die resultierenden Punkte hingegen scheinbar wahllos im Raum verteilt, liegt die Vermutung nahe, dass der optimale Zeitparameter überschätzt wurde.

7.1.2 Renyi-Dimensionsschätzer am Beispiel des Henon-Attraktors

Nun wird das Konvergenzverhalten des Boxcounting- und des Korrelationsdimensions-Schätzers untersucht. Als zugrundeliegende Struktur dient uns der chaotische Henon-Attraktor[2] aus Beispiel 2.5 ($a = 1.4$, $b = 0.3$, $\mathbf{z_0} = (0,0)^T$). Die zu approximierende Boxcounting-Dimension ist 1.26, siehe [GWST82]. Der Korrelationsdimensionsschätzer sollte gegen den Wert 1.22 konvergieren, auch wenn die echte Korrelationsdimension nach Unterabschnitt 4.2.3 größer sein muss. Diese Tatsache wird in [GP83] erläutert.
Mittels der Henon-Abbildung (2.21) und der Observablen $o((x_1, x_2)^T) = x_2$ werden die Zeitreihen T_N erzeugt, welche aus $N = 2^i, i = 9, \ldots, 18$ Werten bestehen. Wir verwenden $\tau = 1$ als Delayparameter, da sich durch visuelle Inspektion zeigt, dass der Henon-Attraktor bereits vollständig entfaltet ist. Die Autokorrelationsfunktion bestätigt diese Annahme.
Zunächst werden nun 200 Werte ϵ_i ($S = 10$, $s = 1000000$) wie in (6.2) bestimmt. Für diese werden – analog zu (6.3) – $d_{cap}^{\epsilon_i}$ und $d_{cor}^{\epsilon_i}$ für die Zeitreihe $T_{2^{18}}$ bestimmt. In Abbildung 7.6 sind die entsprechenden Ergebnisse zu sehen.
Nun wird jeweils ein geeigneter x-Achsen-Abschnitt gesucht, in welchem die Schätzun-

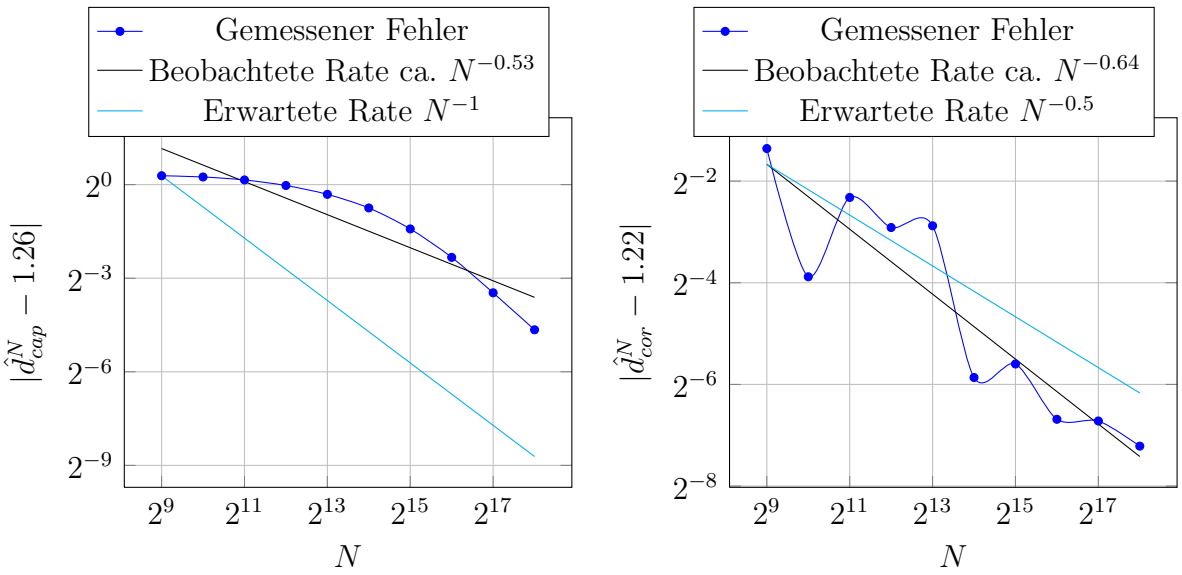
[1] Die Verteilung der Punkte um die Raumdiagonale ist auch für ein größeres Sampling zu beobachten.
[2] Es sei angemerkt, dass das Untersuchen des Lorenz-Attraktors mittels eines Boxcounting-Schätzers zu Problemen führt. Grund hierfür ist die ungleichmäßige Verteilung der Punkte der Trajektorie auf dem Attraktor, siehe [McG83]. Dass diese bei Boxcounting-Schätzern zu ungenauen Schätzungen führen kann, wurde in Unterabschnitt 4.2.6 erläutert.



(a) Boxcounting-Schätzer

(b) Korrelationsdimensionsschätzer

Abb. 7.6: Schätzungen der Renyi-Dimensionen für 200 verschiedene ϵ_i anhand von $T_{2^{18}}$



(a) Boxcounting-Schätzer

(b) Korrelationsdimensionschätzer

Abb. 7.7: Konvergenzverhalten der Renyi-Dimensionsschätzer

gen $d^{\epsilon_i}_{cap}$, bzw. $d^{\epsilon_i}_{cor}$ nahezu identisch sind. Für den Boxcounting-Schätzer schränken wir uns auf den Bereich $[5 \cdot 10^{-4}, 2 \cdot 10^{-3}]$ und für den Korrelationsdimensionsschätzer auf $[10^{-5}, 10^{-3}]$ ein. Für den Boxcounting-Schätzer sei angemerkt, dass der Bereich $\epsilon < 10^{-5}$ keine sinnvolle Wahl wäre. Zwar sind die $d^{\epsilon_i}_{cap}$ dann nahezu identisch, allerdings nur da

2^i	$\lvert\hat{d}^{2^i}_{cap} - 1.26\rvert$	$\frac{\lvert\hat{d}^{2^{i+1}}_{cap} - 1.26\rvert}{\lvert\hat{d}^{2^i}_{cap} - 1.26\rvert}$	$\lvert\hat{d}^{2^i}_{cor} - 1.22\rvert$	$\frac{\lvert\hat{d}^{2^{i+1}}_{cor} - 1.22\rvert}{\lvert\hat{d}^{2^i}_{cor} - 1.22\rvert}$
512	$1.22{\cdot}10^{0}$	1.03	$3.90{\cdot}10^{-1}$	5.75
1,024	$1.19{\cdot}10^{0}$	1.07	$6.78{\cdot}10^{-2}$	0.34
2,048	$1.11{\cdot}10^{0}$	1.13	$2.00{\cdot}10^{-1}$	1.51
4,096	$9.80{\cdot}10^{-1}$	1.22	$1.33{\cdot}10^{-1}$	0.98
8,192	$8.06{\cdot}10^{-1}$	1.35	$1.36{\cdot}10^{-1}$	7.92
16,384	$5.95{\cdot}10^{-1}$	1.59	$1.72{\cdot}10^{-2}$	0.83
32,768	$3.74{\cdot}10^{-1}$	1.88	$2.07{\cdot}10^{-2}$	2.13
65,536	$1.99{\cdot}10^{-1}$	2.20	$9.72{\cdot}10^{-3}$	1.02
131,072	$9.04{\cdot}10^{-2}$	2.27	$9.50{\cdot}10^{-3}$	1.41
262,144	$3.98{\cdot}10^{-2}$	–	$6.74{\cdot}10^{-3}$	–

Tab. 7.8: Fehler der Dimensionsschätzer

die Boxgröße so klein ist, dass disjunkte Punkte oftmals in disjunkten Boxen liegen.
Im entsprechenden Abschnitt werden erneut 200, auf der logarithmischen Skala äquidistante Werte $\tilde{\epsilon}_i$ erzeugt. Für jede Zeitreihe T_N wird nun

$$\hat{d}^N_{cap} := \frac{1}{200} \sum_{i=1}^{200} d^{\tilde{\epsilon}_i}_{cap}$$

als Schätzung verwendet. Analog wird $\hat{d}^N_{cor}$ definiert.
Die resultierenden Fehlerplots sind in Abbildung 7.7 zu sehen. Die "beobachtete" Fehlerrate ist hierbei wieder durch einen linearen Least-Squares-Fit berechnet worden.
Der Korrelationsdimensionsschätzer übertrifft die erwartete Rate, der Boxcounting-Schätzer unterschreitet sie jedoch. Nach einer Phase der langsameren Konvergenz liegt die Konvergenzrate aber auch bei diesem für $N \geq 2^{15}$ nahe der in Unterabschnitt 4.2.2 beschriebenen Rate von $O(N)$.

7.1.3 Vorhersage mit Dünnen Gittern

Um das Konvergenzverhalten des Dünngitter-Algorithmus zu untersuchen, konstruieren wir über die chaotische Henon-Abbildung analog zum vorherigen Unterabschnitt die Zeitreihe T_{10^6} und bilden die rekonstruierten Vektoren im $\mathbb{R}^2$.[3] Wir berechnen die Lösungen von (6.6) und kombinieren diese mit der Kombinationstechnik (6.7) zur Lösung auf dem regulären dünnen Gittern Ω^s_t. Wir variieren dabei das Level $t = 2, \ldots, 10$ und den Regularisierungsparameter $\lambda \in \{10^{-1}, 10^{-2}, \ldots, 10^{-6}\}$.

[3] Es sei angemerkt, dass diese Experimente auch mit T_{10^5} und T_{10^4} durchgeführt wurden, die resultierenden Fehler aber für alle Zeitreihen nahezu identisch waren.

Konvergenz in L_2, L_∞ und H^1, bzw. H^1_{mix}

Zunächst untersuchen wir die Konvergenz der approximierenden Funktionen gegen die H^1-, bzw. H^1_{mix}-regularisierte Lösung $\hat{f}^\lambda_{10}$ auf dem vollen Gitter Ω_{10} des Levels 10. Wir betrachten den L_2- und L_∞-Fehler

$$||e_t||_*$$

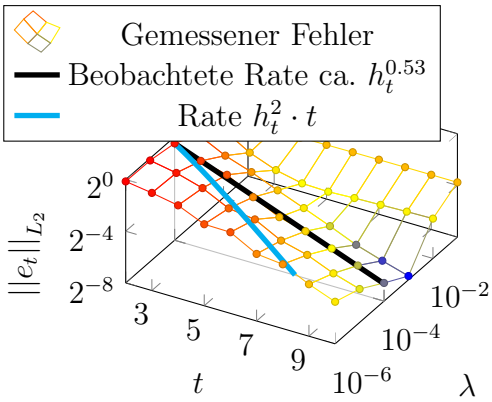
auf dem vollen Gitter Ω_{10}, wobei

$$e_t := f^\lambda_t - \hat{f}^\lambda_{10}$$

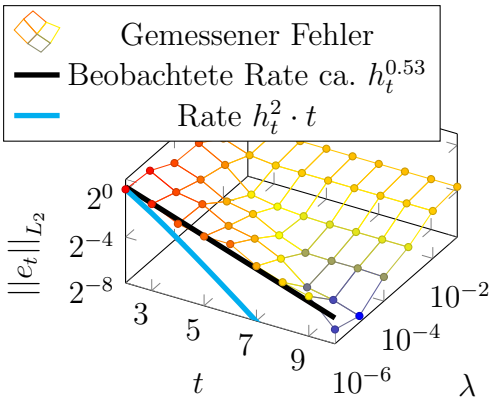
ist und $||\cdot||_*$ die entsprechende Norm darstellt. Zusätzlich betrachten wir den Fehler in der H^1-, bzw. H^1_{mix}-Seminorm. In Abbildung 7.9 und den Tabellen in 7.10 sind die Fehler dargestellt.
Für große λ ist der Fehler für sämtliche Level nahezu gleich. Dies ist auf die Tatsache zurückzuführen, dass die rekonstruierten Funktionen sehr glatt sind und somit eine höhere Auflösung des Gitters nicht zu einer besseren Approximation führt. Für kleine λ lässt sich eine Konvergenz der Dünngitterapproximation erkennen. In Abbildung 7.9 sind verschiedene beobachtete Raten als Least-Squares-Fit in Abhängigkeit von der Maschenweite $h_t := 2^{-t}$ zu sehen.
Die beobachteten Konvergenzraten liegen weit unter der in Abschnitt 5.3 erwähnten Rate $O\left(h_t^2 \cdot t^{d-1}\right)$ für die L_2- und L_∞-Norm, bzw. $O\left(h_t\right)$ für die H^1- und H^1_{mix}-Seminorm. Wie dort bereits erwähnt wurde, gelten diese Raten lediglich für eine Interpolation auf dem gesamten dünnen Gitter. Da wir nicht die Werte an den Dünngitterpunkten, sondern die Trainingsdaten für das zu minimierende Funktional vorgegeben haben, können wir nicht erwarten, dass der Fehler gegen 0 geht.
Die Reduktionsraten des L_2- und H^1-, bzw. H^1_{mix}-Fehlers sind für beide Regularisierungen nahezu identisch. Für den L_∞-Fehler ist die Rate im H^1_{mix}-Fall allerdings besser als im H^1-Fall. Anzumerken ist, dass die Fehlerentwicklung bei einer H^1_{mix}-Regularisierung mit der einer H^1-Regularisierung für ein 100 mal größeres λ vergleichbar ist. Dies ist auf die Tatsache zurückzuführen, dass H^1_{mix}-regularisierte Funktionen stärkere Glattheitseigenschaften aufweisen als H^1-regularisierte.

Konvergenz in l_2, l_∞

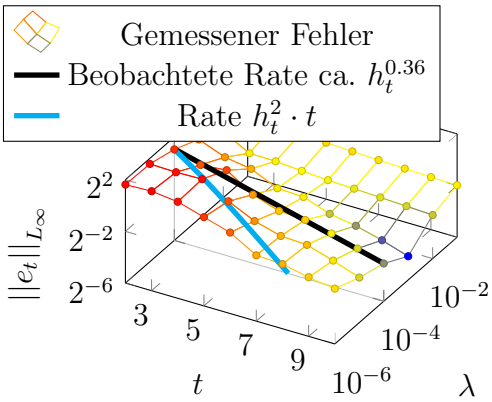
Wie bereits angemerkt, ist der L_2-, bzw. L_∞-Fehler bezüglich des Lebesgue-Maßes kein geeignetes Fehlermaß. Wir wollen nun die L_2- und L_∞-Fehler bezüglich einer empirischen Verteilung auf dem Attraktor messen. Dies entspricht den im Abschnitt 5.3 erwähnten Root-Mean-Squared- (RMSE oder l_2-Fehler) und Maximums-Fehlern (l_∞-Fehler). Um die Testdaten zu erzeugen wurden die Werte $s_{10^6+1}, \ldots, s_{2\cdot 10^6}$ der Henon-Zeitreihe eingebettet. Dies entspricht einer Einbettung der Zeitreihe $T_{2\cdot 10^6}$ ohne die ersten 10^6 Werte.



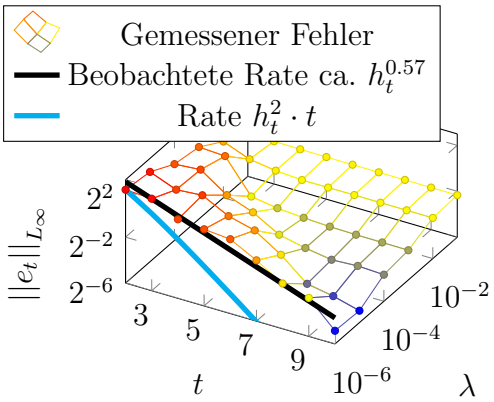
(a) L_2-Fehler, H^1-Regularisierung



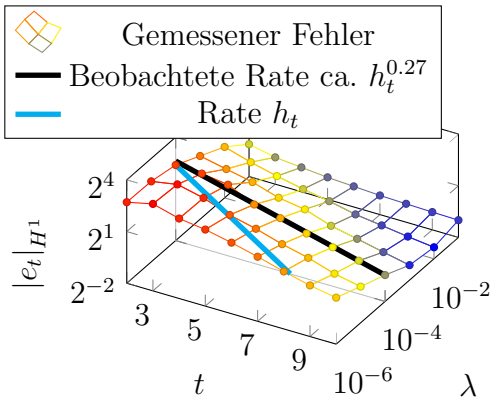
(b) L_2-Fehler, H^1_{mix}-Regularisierung



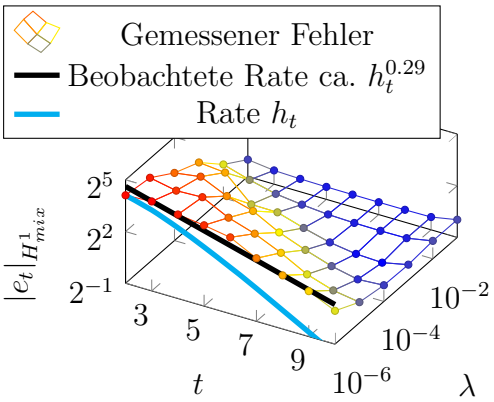
(c) L_∞-Fehler, H^1-Regularisierung



(d) L_∞-Fehler, H^1_{mix}-Regularisierung



(e) H^1-Fehler, H^1-Regularisierung



(f) H^1_{mix}-Fehler, H^1_{mix}-Regularisierung

Abb. 7.9: Konvergenz der Dünngitterapproximation für die Zeitreihe T_{10^6} und beobachtete Raten für $\lambda = 10^{-4}$ für H^1-Regularisierung und $\lambda = 10^{-6}$ für H^1_{mix}-Regularisierung

t	$\|\|e_t\|\|_{L_2}$	$\frac{\|\|e_{t+1}\|\|_{L_2}}{\|\|e_t\|\|_{L_2}}$	$\|\|e_t\|\|_{L_\infty}$	$\frac{\|\|e_{t+1}\|\|_{L_\infty}}{\|\|e_t\|\|_{L_\infty}}$	$\|e_t\|_{H^1_{mix}}$	$\frac{\|e_{t+1}\|_{H^1}}{\|e_t\|_{H^1}}$
2	$7.07{\cdot}10^{-1}$	1.70	$2.06{\cdot}10^{0}$	1.53	$5.22{\cdot}10^{0}$	1.04
3	$4.14{\cdot}10^{-1}$	1.40	$1.34{\cdot}10^{0}$	0.88	$5.02{\cdot}10^{0}$	1.15
4	$2.95{\cdot}10^{-1}$	2.21	$1.53{\cdot}10^{0}$	2.47	$4.36{\cdot}10^{0}$	1.63
5	$1.34{\cdot}10^{-1}$	1.94	$6.19{\cdot}10^{-1}$	1.60	$2.67{\cdot}10^{0}$	1.27
6	$6.89{\cdot}10^{-2}$	1.41	$3.87{\cdot}10^{-1}$	1.08	$2.10{\cdot}10^{0}$	1.37
7	$4.87{\cdot}10^{-2}$	1.68	$3.60{\cdot}10^{-1}$	1.37	$1.53{\cdot}10^{0}$	1.28
8	$2.91{\cdot}10^{-2}$	1.65	$2.62{\cdot}10^{-1}$	1.49	$1.20{\cdot}10^{0}$	1.26
9	$1.76{\cdot}10^{-2}$	1.70	$1.76{\cdot}10^{-1}$	1.49	$9.48{\cdot}10^{-1}$	1.31
10	$1.03{\cdot}10^{-2}$	–	$1.19{\cdot}10^{-1}$	–	$7.23{\cdot}10^{-1}$	–

(a) $\lambda = 10^{-4}$, H^1-Regularisierung

t	$\|\|e_t\|\|_{L_2}$	$\frac{\|\|e_{t+1}\|\|_{L_2}}{\|\|e_t\|\|_{L_2}}$	$\|\|e_t\|\|_{L_\infty}$	$\frac{\|\|e_{t+1}\|\|_{L_\infty}}{\|\|e_t\|\|_{L_\infty}}$	$\|e_t\|_{H^1_{mix}}$	$\frac{\|e_{t+1}\|_{H^1_{mix}}}{\|e_t\|_{H^1_{mix}}}$
2	$1.11{\cdot}10^{0}$	1.53	$4.38{\cdot}10^{0}$	1.09	$1.72{\cdot}10^{1}$	0.94
3	$7.23{\cdot}10^{-1}$	2.08	$4.02{\cdot}10^{0}$	2.19	$1.83{\cdot}10^{1}$	1.25
4	$3.47{\cdot}10^{-1}$	1.35	$1.84{\cdot}10^{0}$	1.15	$1.47{\cdot}10^{1}$	1.14
5	$2.56{\cdot}10^{-1}$	1.00	$1.60{\cdot}10^{0}$	1.15	$1.29{\cdot}10^{1}$	1.20
6	$2.56{\cdot}10^{-1}$	2.38	$1.39{\cdot}10^{0}$	2.26	$1.08{\cdot}10^{1}$	1.70
7	$1.07{\cdot}10^{-1}$	1.67	$6.15{\cdot}10^{-1}$	2.56	$6.32{\cdot}10^{0}$	1.47
8	$6.42{\cdot}10^{-2}$	1.47	$2.40{\cdot}10^{-1}$	1.50	$4.31{\cdot}10^{0}$	1.39
9	$4.37{\cdot}10^{-2}$	4.70	$1.60{\cdot}10^{-1}$	4.60	$3.10{\cdot}10^{0}$	1.62
10	$9.29{\cdot}10^{-3}$	–	$3.47{\cdot}10^{-2}$	–	$1.91{\cdot}10^{0}$	–

(b) $\lambda = 10^{-6}$, H^1_{mix}-Regularisierung

Tab. 7.10: Tabelle der der L_2-, L_∞- und H^1-, bzw. H^1_{mix}-Fehler

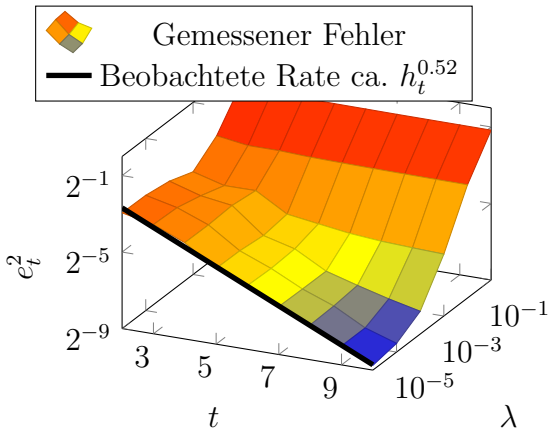
Wir erhalten somit den l_2-Fehler als

$$e_t^2 := \sqrt{\frac{1}{10^6} \sum_{i=10^6+1}^{2{\cdot}10^6} \left(f_t^\lambda(\mathbf{x}_i) - s_{i+1} \right)^2}$$

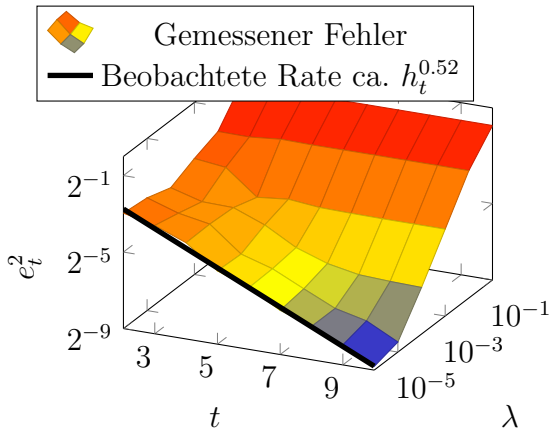
und den l_∞-Fehler als

$$e_t^\infty := \max_{i=10^6+1,\ldots,2{\cdot}10^6} \left| f_t^\lambda(\mathbf{x}_i) - s_{i+1} \right|.$$

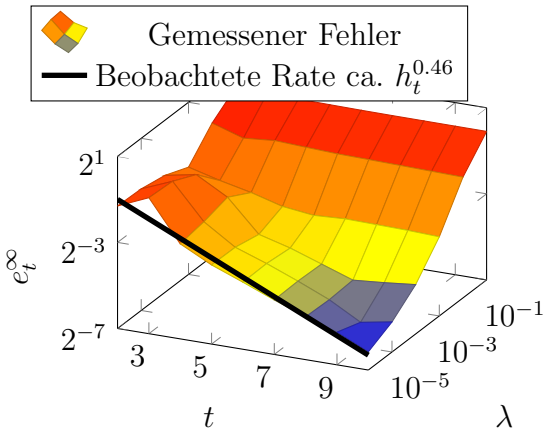
Die resultierenden Raten sind in Abbildung 7.11 zu sehen.
Die zu beobachtenden Konvergenzraten sind ungefähr von der Ordnung $O(\sqrt{h_t})$. Die Qualität der Vorhersage auf den Testdaten wird insbesondere auf den anwendungsorientierten Datensätzen zu überprüfen sein.



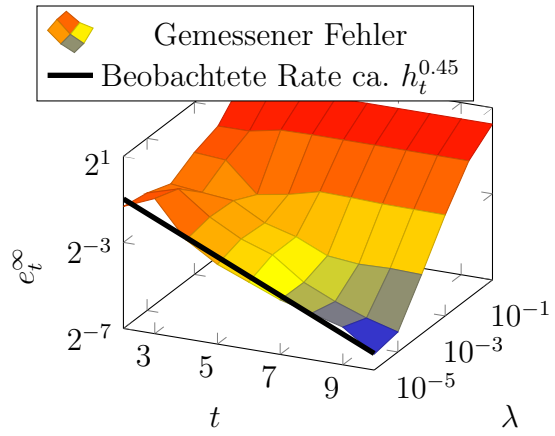
(a) l_2-Fehler, H^1-Regularisierung



(b) l_2-Fehler, H^1_{mix}-Regularisierung



(c) l_∞-Fehler, H^1-Regularisierung



(d) l_∞-Fehler, H^1_{mix}-Regularisierung

Abb. 7.11: Konvergenz der Dünngitterapproximation auf den Testdaten für die Zeitreihe T_{10^6} und beobachtete Raten für $\lambda = 10^{-6}$

Es ist erwähnenswert, dass die Raten für kleinere λ immer besser werden. Dies ist vor allem darauf zurückzuführen, dass der Attraktor eine lokale Struktur besitzt und sich die gesamten Trainings- und Testdaten auf dieser befinden. In der Praxis ist eine Regularisierung dennoch wichtig, da die vorliegenden Prozesse oftmals leicht nichtstationär sind oder die Samplingrate auf dem Attraktor ungenügende Genauigkeit hat. Zudem zeigt ein Experiment mit $\lambda = 0$, dass die Konvergenzraten und die Fehler ohne Regularisierung schlechter sind als für $\lambda = 10^{-6}$. [4]

[4] Der l_2-Fehler auf Level 10 ist mit 0.0027 ohne Regularisierung und der Rate $h_t^{0.51}$ geringfügig schlechter als für den Fall $\lambda = 10^{-6}$. Für den l_∞-Fehler wirkt sich dies mit einem Fehler von 0.037 und einer Rate von $h_t^{-0.34}$ ohne Regularisierung stärker aus. Hier ist der Fall $\lambda = 10^{-6}$ deutlich besser.

t	e_t^2	$\frac{e_{t+1}^2}{e_t^2}$	e_t^∞	$\frac{e_{t+1}^\infty}{e_t^\infty}$
2	$1.30{\cdot}10^{-1}$	1.27	$4.05{\cdot}10^{-1}$	0.65
3	$1.02{\cdot}10^{-1}$	1.60	$6.24{\cdot}10^{-1}$	3.74
4	$6.40{\cdot}10^{-2}$	2.11	$1.67{\cdot}10^{-1}$	1.67
5	$3.03{\cdot}10^{-2}$	1.54	$9.99{\cdot}10^{-2}$	1.48
6	$1.97{\cdot}10^{-2}$	1.77	$6.77{\cdot}10^{-2}$	1.50
7	$1.11{\cdot}10^{-2}$	1.61	$4.52{\cdot}10^{-2}$	1.46
8	$6.88{\cdot}10^{-3}$	1.77	$3.10{\cdot}10^{-2}$	1.39
9	$3.88{\cdot}10^{-3}$	1.67	$2.23{\cdot}10^{-2}$	1.59
10	$2.32{\cdot}10^{-3}$	–	$1.41{\cdot}10^{-2}$	–

(a) H^1-Regularisierung

t	e_t^2	$\frac{e_{t+1}^2}{e_t^2}$	e_t^∞	$\frac{e_{t+1}^\infty}{e_t^\infty}$
2	$1.28{\cdot}10^{-1}$	1.30	$3.98{\cdot}10^{-1}$	0.66
3	$9.88{\cdot}10^{-2}$	1.55	$6.07{\cdot}10^{-1}$	3.21
4	$6.38{\cdot}10^{-2}$	2.29	$1.89{\cdot}10^{-1}$	1.86
5	$2.78{\cdot}10^{-2}$	1.53	$1.02{\cdot}10^{-1}$	1.50
6	$1.82{\cdot}10^{-2}$	1.88	$6.78{\cdot}10^{-2}$	1.50
7	$9.68{\cdot}10^{-3}$	1.48	$4.53{\cdot}10^{-2}$	1.30
8	$6.53{\cdot}10^{-3}$	1.62	$3.47{\cdot}10^{-2}$	1.32
9	$4.04{\cdot}10^{-3}$	1.81	$2.63{\cdot}10^{-2}$	2.02
10	$2.23{\cdot}10^{-3}$	–	$1.30{\cdot}10^{-2}$	–

(b) H^1_{mix}-Regularisierung

Tab. 7.12: Tabelle der Fehler für $\lambda = 10^{-6}$

7.2 Dimensionsschätzungen in großen Dimensionen

7.2.1 Concentration of Measure

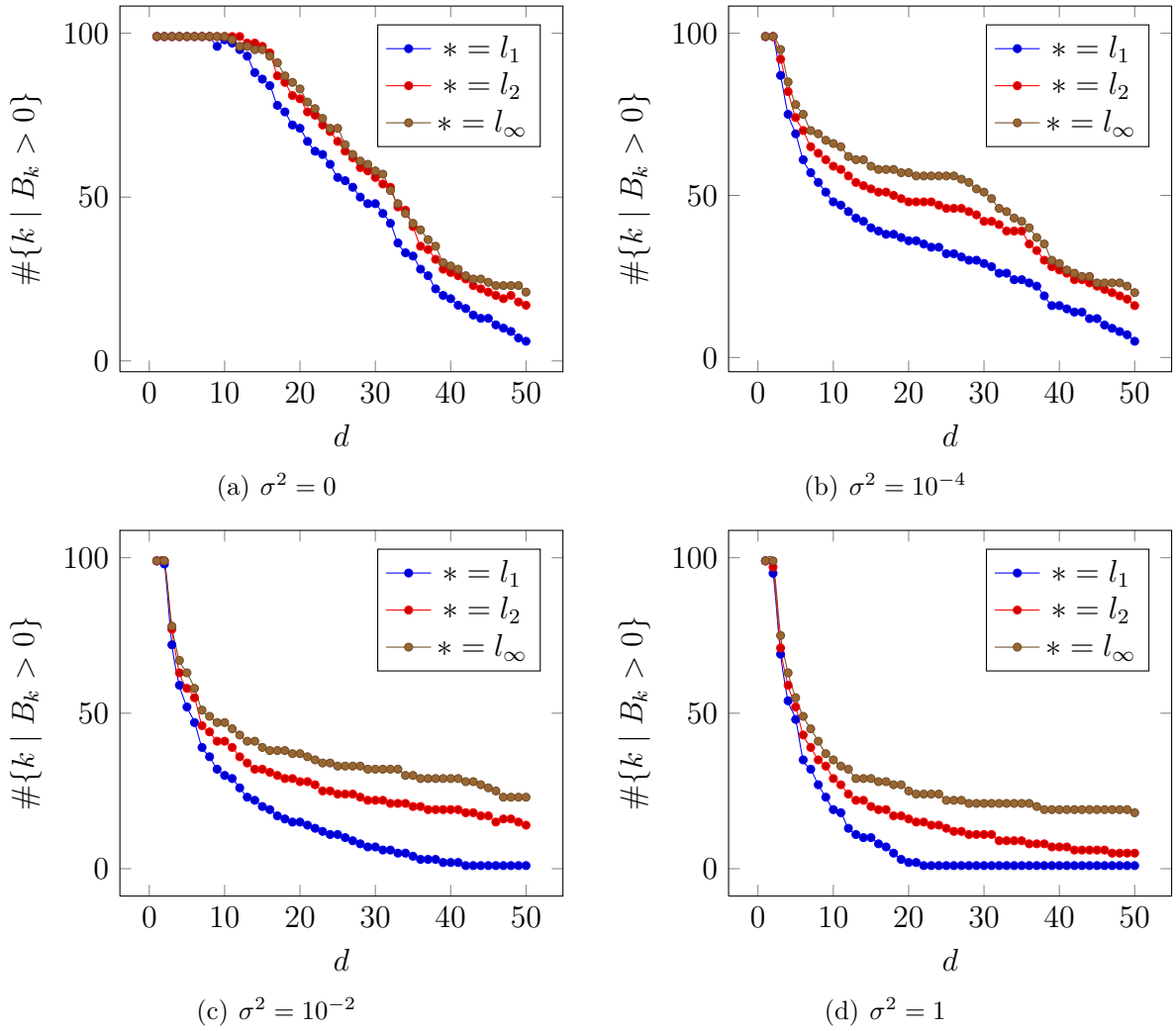
In Unterabschnitt 4.2.7 wurden der Concentration of Measure-Effekt und seine Folgen für Abstandsmessungen in großen Dimensionen d erläutert. Am Beispiel der Anwendung des Korrelationsdimensionsschätzers auf die chaotische Henon-Abbildung wollen wir die Auswirkungen dieses Phänomens betrachten. Analog zu den vorherigen Experimenten wird die Zeitreihe T_{10^4} aus der Henon-Abbildung konstruiert. Jeder Zeitreihenpunkt wird dann mit der Normalverteilung $N(0, \sigma^2)$ verrauscht. Im Anschluss wird die Zeitreihe in verschiedene Dimensionen eingebettet. Auf die eingebetteten Punkte wird der Korrelationsdimensionsalgorithmus angewendet. Als Normen im Argument der Heavyside-Funktion

$$H(\epsilon - ||\mathbf{x} - \mathbf{y}||_*)$$

werden $* = l_1, l_2, l_\infty$ verwendet. Die Korrelationssummen werden für 100 logarithmisch äquidistante Werte ϵ_i zwischen 1 und 10^{-4} mittels des in Unterabschnitt 6.1.2 beschriebenen Histogramm-Ansatz (6.5) berechnet. Für eine gute Schätzung ist es notwendig, dass viele der Histogramm-Einträge

$$B_k := \frac{1}{2}\#\left\{(\mathbf{x_i}, \mathbf{x_j}) \mid \epsilon_k \geq ||\mathbf{x_i} - \mathbf{x_j}||_2 > \epsilon_{k+1}\right\}$$

nicht Null sind. In Abbildung 7.13 ist die Anzahl der Nicht-Null-Histogramm-Einträge für verschiedene Einbettungsdimensionen d, Normen $||\cdot||_*$ und Varianzen σ^2 zu sehen. Dass die l_∞ und l_2-Norm in hohen Dimensionen wirklich bessere Schätzungen für die Korrelationsdimension liefern können als die l_1-Norm, ist in Abbildung 7.15 zu sehen.



(a) $\sigma^2 = 0$ (b) $\sigma^2 = 10^{-4}$

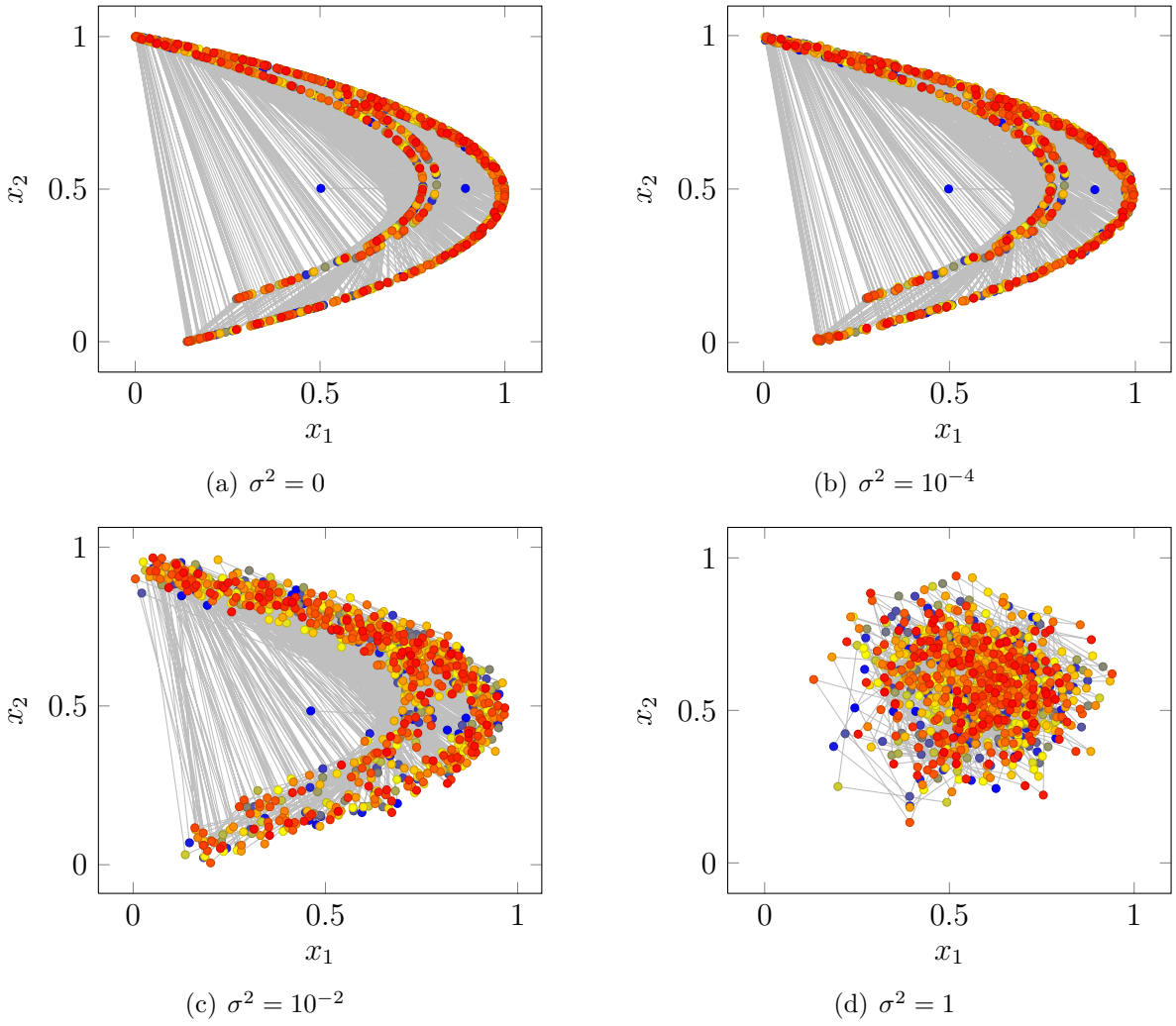
(c) $\sigma^2 = 10^{-2}$ (d) $\sigma^2 = 1$

Abb. 7.13: Anzahl der Nicht-Null-Histogramm-Einträge in Abhängigkeit der Dimension d, der Norm $||\cdot||_*$ und der Varianz σ^2

Je größer σ^2 wird, umso stärker ist der Concentration of Measure-Effekt zu sehen: Die Abstände zwischen beliebigen Punkten sind in hohen Dimensionen fast gleich und nur noch wenige B_k sind echt positiv. Zu sehen ist, dass dieser Effekt für die l_1-Norm stärker ist als für die l_2- bzw. l_∞-Norm. Dies scheint auf den ersten Blick nicht das zu erwartende Verhalten zu sein, da wir sehen, dass die l_∞-Norm in diesem Beispiel das Distanzmaß ist, für welches

$$\overline{\mathcal{D}} := \left| \max_{\mathbf{x}} ||\mathbf{x}||_p - \min_{\mathbf{x}} ||\mathbf{x}||_p \right|$$

in allen Dimensionen am größten ist. Dies ist jedoch kein Widerspruch zu den in Unterabschnitt 4.2.7 präsentierten Ergebnissen. Dieses Verhalten entsteht durch die Ska-



(a) $\sigma^2 = 0$ (b) $\sigma^2 = 10^{-4}$

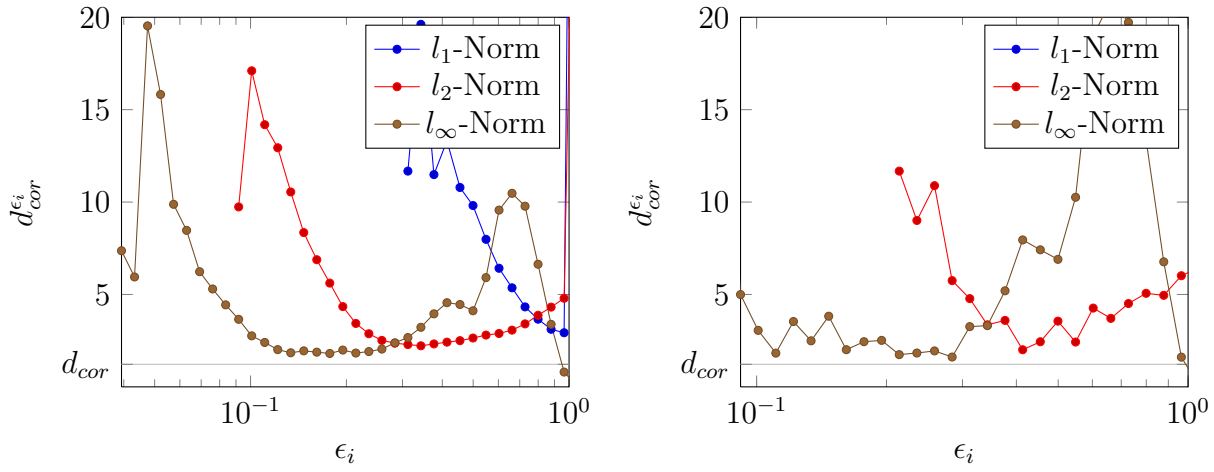
(c) $\sigma^2 = 10^{-2}$ (d) $\sigma^2 = 1$

Abb. 7.14: Die ersten 1000 Punkte der Rekonstruktionen des $N(0, \sigma^2)$-verrauschten Henon-Attraktors im $\mathbb{R}^2$

lierung der Zeitreihe auf $[0, 1]$. In [HAK00] wird erwähnt, dass $\min_{\mathbf{x},\mathbf{y}} ||\mathbf{x} - \mathbf{y}||_p$ und $\max_{\mathbf{x},\mathbf{y}} ||\mathbf{x} - \mathbf{y}||_p$ in einem Datensatz aus Zufallszahlen für kleine p wesentlich schneller wachsen als für große p. Zusätzlich gilt allerdings

$$\frac{\max_{\mathbf{x},\mathbf{y}} ||\mathbf{x} - \mathbf{y}||_p - \min_{\mathbf{x},\mathbf{y}} ||\mathbf{x} - \mathbf{y}||_p}{\min_{\mathbf{x},\mathbf{y}} ||\mathbf{x} - \mathbf{y}||_p} \to 0 \quad \forall\, p.$$

Dies liefert die Erklärung für das beobachtete Verhalten der l_1-Norm: Durch die Skalierung der Punkte sind alle Abstände bereits in kleineren Dimensionen nahe bei 1 als dies für die l_∞-Norm der Fall ist.
Das hier beobachtete Phänomen ist eng verwandt mit einem weiteren Concentration of



(a) $d = 20$ (b) $d = 40$

Abb. 7.15: Schätzungen der Korrelationsdimension des Henon-Attraktors für $\sigma = 10^{-2}$

Measure-Effekt, welcher dazu führt, dass sich Zufallszahlen aus $[0, 1]^d$ für große d in den Ecken des Hyperwürfels konzentrieren, siehe [LV07].
Zusammengefasst ist somit zu sehen, dass Minkowski-Normen mit großem p in diesem Fall ein sinnvolleres Distanzmaß als Minkowski-Normen mit kleinem p darstellen.
Es ist auch anzumerken, dass der Concentration of Measure-Effekt – wenn auch in schwächerer Form – auch für den unverrauschten Prozess mit $\sigma^2 = 0$ eintritt. Außerdem ist es bemerkenswert, wie stark der Abfall von $\#\{k \mid B_k > 0\}$ in den ersten 10 Dimensionen bereits für $\sigma^2 = 10^{-4}$ ist. Diese minimale Störung der Zeitreihendaten ist in 2 Dimensionen kaum sichtbar – siehe Abbildung 7.14 – und hat dennoch bereits in kleinen Dimensionen erhebliche Auswirkungen auf die Anzahl der echt positiven Histogramm-Einträge.
Für $d = 20$ ist für den l_2- und l_∞-Schätzer ein deutliches Plateau im Bereich der echten Korrelationsdimension d_{cor} zu sehen. Der l_1-Schätzer liefert hier bereits keine verlässlichen Resultate mehr. Für $d = 40$ liegen sämtliche l_1-Abstände bereits so nahe bei 1, dass nur noch ein B_k ungleich 0 ist. Der l_2-Schätzer liefert kein Plateau in der Nähe von d_{cor}. Der l_∞-Schätzer liefert zwar auch kein deutliches Plateau, es ist aber eine Tendenz erkennbar.

7.2.2 Clustering mit Bregman-Divergenzen

Um die Auswirkungen verschiedener Bregman-Divergenzen im Clustering-Algorithmus zu beobachten, verwenden wir eine Trajektorie auf der eindimensionalen Sphäre S^1 mit Radius $r = 20$. Den korrespondierenden konservativen zeitdiskreten Prozess auf der Sphäre definieren wir durch den Diffeomorphismus $\phi : S^1 \to S^1$:

$$\phi\left(r \cdot \exp(i\theta)\right) = r \cdot \exp(i(\theta + 1))$$

Als Anfangswert wird $\theta_0 = 0$ verwendet. Mittels der Observable

$$o((r \cdot \exp(i\theta))) := \Re\,(r \cdot \exp(i\theta)) = r \cdot \cos(\theta)$$

erzeugen wir die Zeitreihe T_{10^5}. Diese wird anschließend $N(0, \sigma^2)$-verrauscht und in $\mathbb{R}^{20}$ eingebettet. Wir stellen im Folgenden vier Möglichkeiten vor, um mittels der lokalen PCA die Dimension der Sphäre zu schätzen:

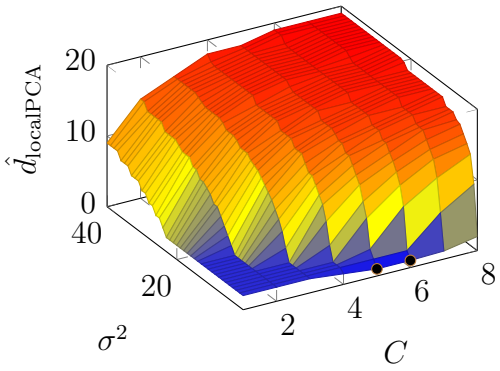
1. Wir führen ein Clustering mit der quadrierten euklidischen Norm als Distanzmaß durch und schätzen anschließend die Dimension der verrauschten Sphäre mit einer lokalen PCA mit 95% Varianzerhaltung in jedem Cluster. In Abbildung 7.16 (a) sind die geschätzten Dimensionen $\hat{d}_{\text{localPCA}}$ als arithmetisches Mittel der PCA-Dimensionen der einzelnen Cluster in Abhängigkeit von der Clusteranzahl C und der Varianz σ^2 zu sehen. Die lokale PCA für $C = 1$ entspricht hierbei der gewöhnlichen PCA. Diese kann bestenfalls $\hat{d}_{\text{PCA}} = 2$ liefern, da die Sphäre mit linearen Verfahren nicht in $\mathbb{R}^1$ eingebettet werden kann.

2. Eine andere Variante zur Schätzung ergibt sich, wenn die Daten zunächst auf die ersten $\hat{d}_{\text{PCA}}$ Hauptachsen projeziert werden und auf den resultierenden Vektoren eine lokale PCA durchgeführt wird. Um einen Vergleich zum ersten Schätzer zu ermöglichen, wurden sowohl die PCA als auch die darauffolgende lokale PCA mit einem Varianzerhalt von $\sqrt{0.95}\cdot 100\%$ durchgeführt. Das Ergebnis der lokalen PCA-Schätzung ist in Abbildung 7.16 (b) zu sehen.

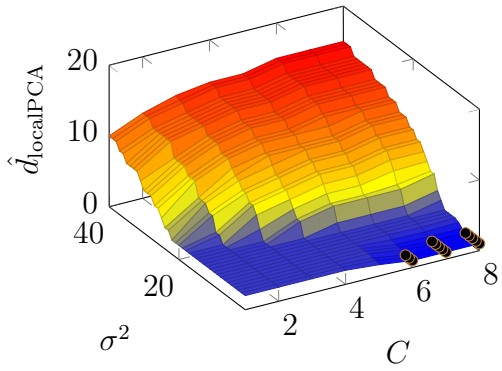
3. Eine dritte Variante erhält man dadurch, dass man für alle Punkte in $\mathbb{R}^{20}$ ein Clustering mit der quadrierten euklidischen Norm und der Clusteranzahl 10^3 durchführt und anschließend eine lokale PCA mit 95% Varianzerhalt auf der Menge der 10^3 gefundenen Clusterpunkte (*Repräsentanten*) durchführt. Mit dieser Methode kann die Dimension des Objektes theoretisch unterschätzt werden. Liegen allerdings genügend Repräsentanten vor und stellen diese die ursprüngliche Struktur genügend gut dar, so liefert dies eine gute Alternative zu den ersten beiden Methoden. Die Resultate dieser Methode sind in Abbildung 7.16 (c) zu finden.

4. Eine Kombination der vorgestellten Methoden ergibt folgende Strategie: Zunächst wird eine globale PCA mit $\sqrt{0.95} \cdot 100\%$ Varianzerhaltung durchgeführt. Auf der Menge der transformierten Punkte wird anschließend ein Clustering mit der quadrierten euklidischen Norm und 10^3 Clusterpunkten durchgeführt. Auf den resultierenden 10^3 Repräsentanten wird der Algorithmus der lokalen PCA mit $\sqrt{0.95}\cdot 100\%$ Varianzerhalt ausgeführt. Die Ergebnisse sind in 7.16 (d) zu sehen.

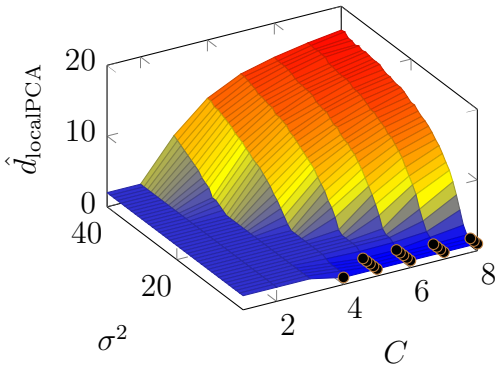
In diesem Beispiel erhält man den Wert 1 als Schätzung $\hat{d}_{\text{localPCA}}$ nur für solche Parameter (σ^2, C), für welche die PCA mit 95% Varianzerhalt bereits die geschätzte Dimension 2 liefert. Somit ist es hier eigentlich nicht sinnvoll die lokale PCA als Referenz zu nehmen, da man nach Takens selbst bei der Schätzung der korrekten Dimension bereits in $\mathbb{R}^3$



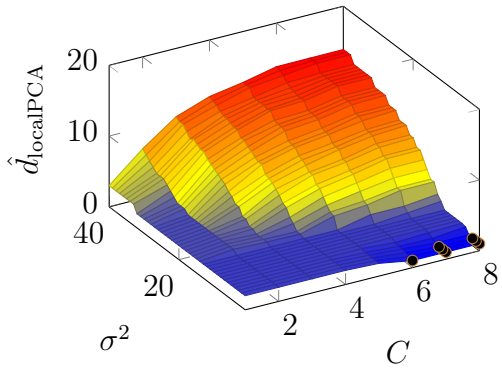
(a) lokale PCA



(b) PCA → lokale PCA



(c) Clustering → lokale PCA auf 10^3 Repräsentanten



(d) PCA → Clustering → lokale PCA auf 10^3 Repräsentanten

Abb. 7.16: Resultate des lokalen PCA-Schätzers in Abhängigkeit von C und σ^2 – Ein Kreis markiert die Parametertupel (σ^2, C), für welche der Schätzer exakt den korrekten Wert 1 liefert.

einbetten müsste. Wir ziehen dieses Beispiel dennoch heran, um zu verdeutlichen, wie sich verschiedene Bregman-Divegenzen auf das Clustering und die Dimensionsschätzung auswirken.
Um diesen Effekt zu zeigen, wählen wir die dritte beschriebene Variante. Diese liefert für $\sigma = 5$ ($C = 5, 6$) noch die korrekte Dimension. Ebenfalls liefert die gewöhnliche PCA auf den errechneten 10^3 Repräsentanten selbst für $\sigma = 40$ noch die bestmögliche Schätzung 2.
Wir verrauschen nun die ursprüngliche Zeitreihe mit einer Standardnormalverteilung

$$\underbrace{f(x)}_{\text{Dichte}} = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-x^2}{2\sigma^2}\right),$$

einer Exponentialverteilung

$$\underbrace{f(x)}_{\text{Dichte}} = \frac{1}{\sigma} \exp\left(-\frac{1}{\sigma}x\right)$$

und einer diskreten Poissonverteilung

$$\underbrace{p(k)}_{\text{Wahrscheinlichkeit}} = \frac{\sigma^{2k}}{k!} \exp(-\sigma^2).$$

Als Abstandsmaße beim Clustering für die lokale PCA sowie bei der Repräsentantensuche wählen wir die quadrierte euklidische Norm

$$d(\mathbf{x}, \mathbf{y}) = ||\mathbf{x} - \mathbf{y}||_2^2,$$

die *Itakura-Saito-Divergenz*

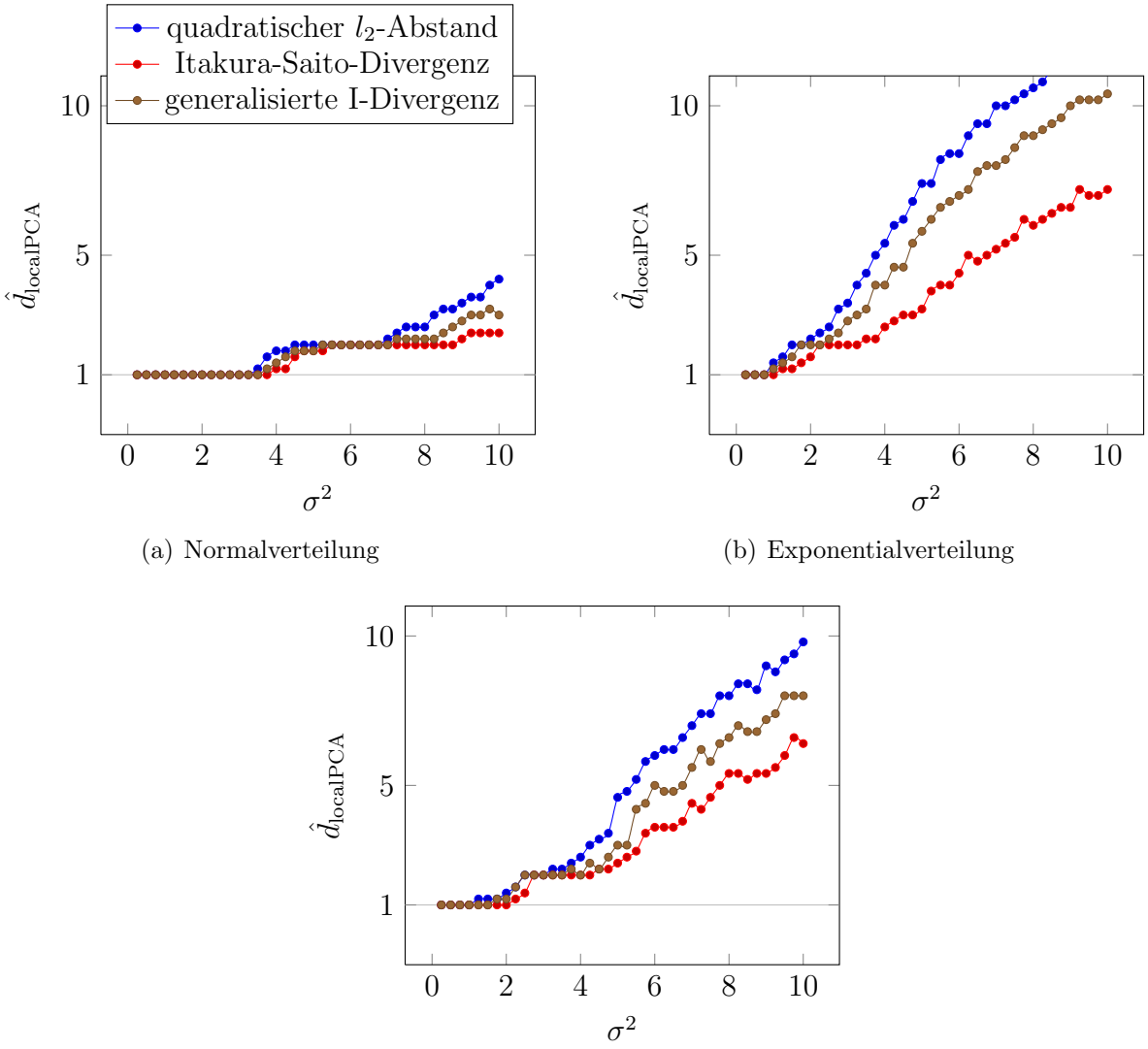
$$d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{d} \frac{x_i}{y_i} - \log\left(\frac{x_i}{y_i}\right) - 1$$

und die *generalisierte Kullback-Leibler-Divergenz* (auch *generalisierte I-Divergenz*)

$$d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{d} x_i \cdot \log\left(\frac{x_i}{y_i}\right) - (x_i - y_i).$$

Wie man in Tabelle 4.2 sehen kann, sind dies gerade die zu den Verteilungen korrespondierenden Bregman-Divergenzen. [5] Die folgenden Kalkulationen führen wir auf der verkürzten Zeitreihe T_{10^4} durch. Wir berechnen nun die Resultate des lokalen PCA-Schätzers für $\sigma^2 = 1, \ldots, 40$ mit $C = 5$ auf 500 vorher errechneten Repräsentanten im $\mathbb{R}^{20}$ für alle neun Kombinationen aus Wahrscheinlichkeitsverteilungen und Bregman-Divergenzen. Die Länge der Zeitreihe sowie die Anzahl der Repräsentanten wurden reduziert, da die Auswertung des Logarithmus beim Berechnen der Itakura-Saito- und der generalisierten I-Divergenz viel Rechenzeit in Anspruch nimmt. Die Ergebnisse sind in Abbildung 7.17 zu sehen.
Ein direkter Zusammenhang zwischen der zugrundeliegenden Verteilung und der korrespondierenden Bregman-Divergenz ist nur schwer zu erkennen. Man sieht jedoch, dass die Divergenzen, welche zum entsprechenden Rauschen gehören, bis $\sigma^2 \approx 4$ mindestens so gut abschneiden wie die anderen Abstandsmaße. Allerdings ist insgesamt zu beobachten, dass die Itakura-Saito-Divergenz für alle drei Verteilungen am besten abschneidet. Für die zur Itakura-Saito-Divergenz korrespondierende Exponentialverteilung ist der Ab-

[5] Um insbesondere die Itakura-Saito-Divergenz auf den Datenpunkten auswerten zu können, müssen diese in $\mathbb{R}^+ \setminus \{0\}$ liegen. Aus diesem Grund wurde für dieses Experiment eine Skalierung auf $[0.1, 0.9]^d$ statt $[0, 1]^d$ vorgenommen.



(a) Normalverteilung

(b) Exponentialverteilung

(c) Poissonverteilung

Abb. 7.17: Geschätzte Dimension der verrauschten Sphäre mittels einer lokalen PCA mit fünf Clustern anhand von 500 vorher berechneten Repräsentanten

stand zwischen den geschätzten Dimensionen am deutlichsten.[6] Die Tatsache, dass die geschätzten Dimensionen für gleiche Varianzen bei normalverteiltem Rauschen deutlich besser sind als bei den anderen Verteilungen, ist darauf zurückzuführen, dass das PCA-Modell gerade auf der Annahme von normalverteiltem Rauschen basiert. Einen Ansatz

[6]Bei einer Durchführung des gleichen Experiments in Dimension 5 stellt man ähnliche Resultate fest. Hier ist das bessere Abschneiden der Itakura-Saito-Divergenz gegenüber den anderen beiden Abstandsmaßen allerdings nicht ganz so deutlich.

für eine PCA, die auf anderen Verteilungen basiert, ist in [CDS01] zu finden.
In unserem Experiment überwiegt offenbar der Einfluss, den die Gestalt der eingebetteten Struktur auf die Clustering-Distanz ausübt. In [BMDG05] wird erwähnt, dass bei Kenntnis der zugrundeliegenden Verteilung die Wahl der korrespondierenden Bregman-Divergenz nicht unbedingt die beste sein muss. Schreibt man einen der in $\mathbb{R}^{20}$ eingebetteten Vektoren aus, so hat dieser die Gestalt

$$\mathbf{x}_n = r \cdot (\cos(\theta_0 + n + 19), \ldots, \cos(\theta_0 + n))^T.$$

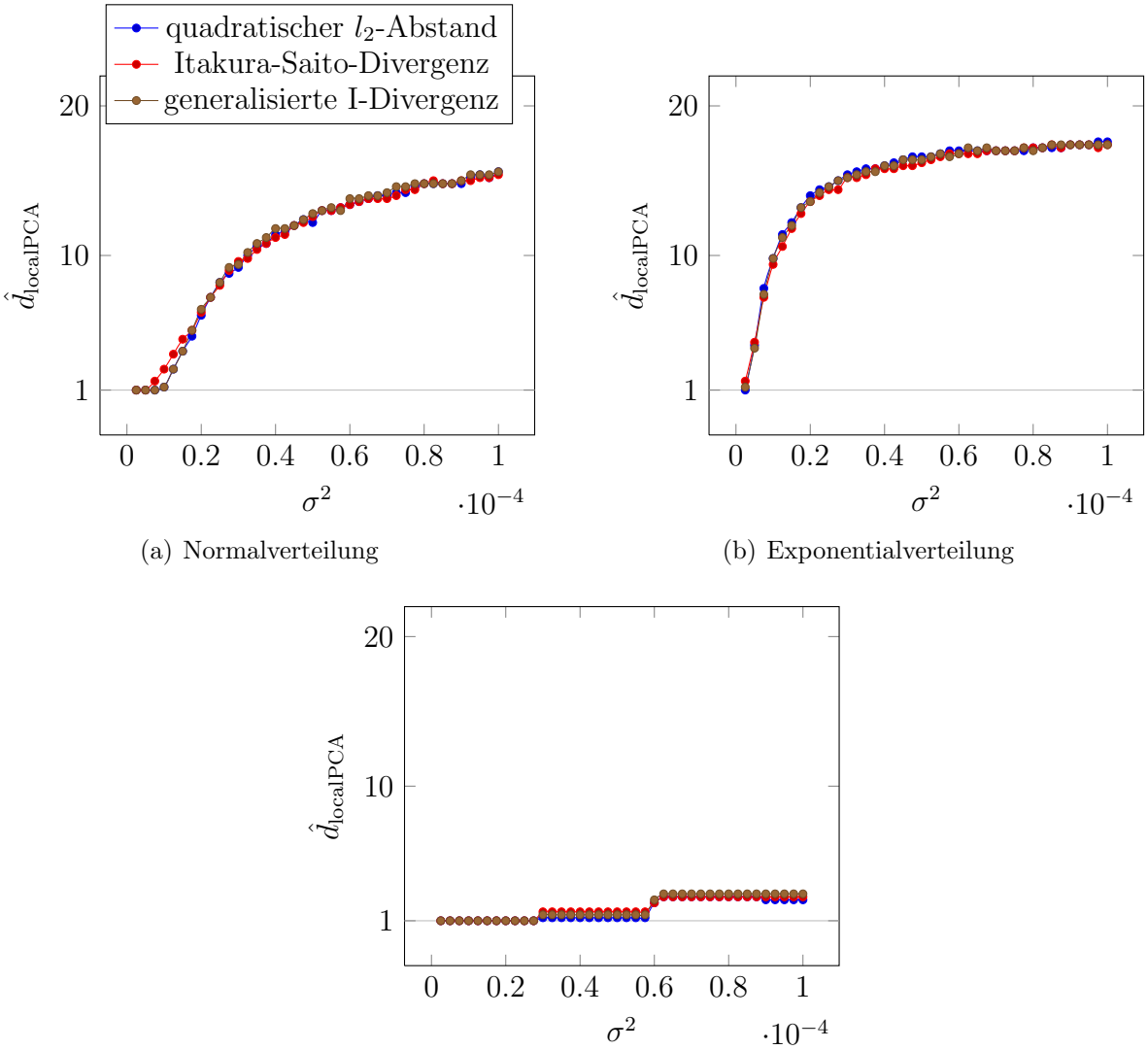
Dies ist ein bestimmter Ausschnitt der diskreten *Laplace-Transformation* des konstanten Radius r:

$$\hat{X}(\psi + i\omega) = \sum_{k=0}^{\infty} r \cdot \exp(-(\psi + i\omega) \cdot (t_0 + k)) = \frac{r}{\psi + i\omega}$$

Für $\psi = 0$, $\omega = -1$, $t_0 = \theta_0$, können die Koeffizienten des eingebetteten Vektors als Realteile von den aufeinanderfolgenden Summanden für $k = n, \ldots, n+19$ in der Laplace-Transformation angesehen werden. Somit kann die in einem Vektor enthaltene Information als eine Approximation des Spektrums der komplexen Frequenz $-i$ angesehen werden, indem alle restlichen Summanden vernachlässigt werden. Analog kann der j-te Koeffizient von $\mathbf{x}_n$ als der n-te Summand in der Laplace-Transformation von $r \cdot \exp(i \cdot (20 - j))$ aufgefasst werden.
Diese Betrachtungsweise liefert eine mögliche Begründung für das gute Abschneiden der Itakura-Saito-Divergenz, die sich vor allem als Distanzmaß in der Spektralanalyse von Signalen bewährt hat, siehe [BMDG05, LBG80].
Dass die Itakura-Saito-Divergenz nicht grundsätzlich besser abschneidet als die anderen Abstandsmaße, zeigt das folgende Beispiel: Wir definieren einen eindimensionalen Prozess durch

$$\phi(x) = x + 10^{-4} \mod 1$$

und erzeugen mittels $o \equiv \mathrm{id}$ und $x_0 = 0.5$ die Zeitreihe T_{10^4}. Die Punkte des eingebetteten Prozesses liegen nahe der Raumdiagonalen und stellen somit approximativ eine eindimensionale Linie dar. Wir führen das gleiche Experiment wie in Abbildung 7.17 im $\mathbb{R}^{20}$ für diese Zeitreihe durch. Die Ergebnisse sind in 7.18 dargestellt.
Das Verhalten der Schätzer für die Poisson-Verteilung ist auf den diskreten Wertebereich der Zufallsvariable zurückzuführen. Insgesamt ist zu sehen, dass keines der drei Abstandsmaße signifikant besser abschneidet als die anderen beiden.
Die Resultate der Experimente zeigen, dass es bei Vorkenntnissen über die Gestalt und die Herkunft des zugrundeliegenden Prozesses sinnvoll sein kann, ein Distanzmaß zu verwenden, welches an diese Rahmenbedingungen angepasst ist. Liegen jedoch keine Vorkenntnisse über den Prozess, sondern ausschließlich über das Rauschen vor, so ist die Wahl der zur entsprechenden Verteilung korrespondierenden Bregman-Divergenz aufgrund der in Unterabschnitt 4.2.7 beschriebenen Zusammenhänge am sinnvollsten.



(a) Normalverteilung

(b) Exponentialverteilung

(c) Poissonverteilung

Abb. 7.18: Geschätzte Dimension der verrauschten Diagonale mittels einer lokalen PCA mit fünf Clustern anhand von 500 vorher berechneten Clusterpunkten

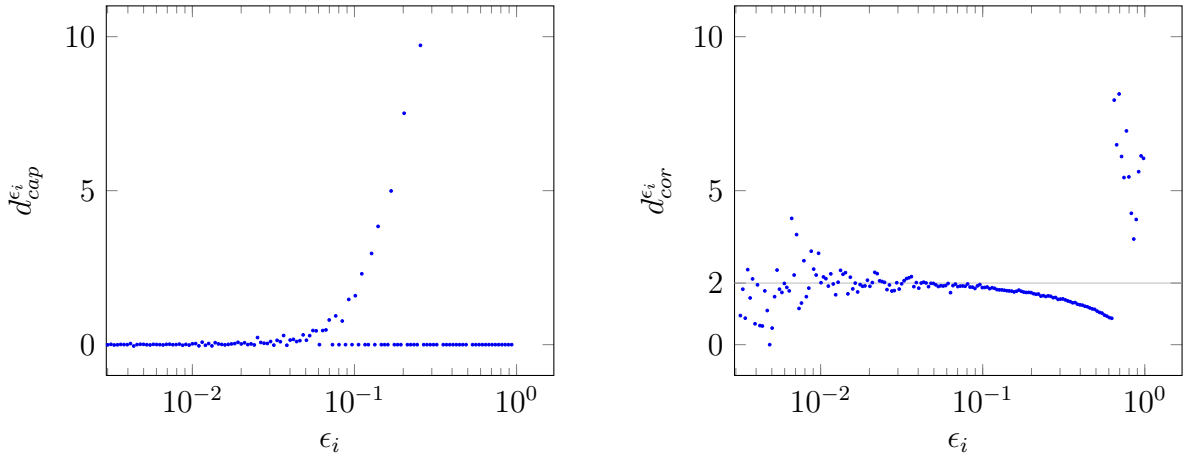
7.3 Kreuzvalidierung und adaptive dünne Gitter

Um die Vorhersage der Zeitreihe mittels einer adaptiven Dünngittermethode zu testen, verwenden das Beispiel einer zweidimensionalen Abbildung, die sich an der eindimensionalen *dyadischen Abbildung* orientiert.

Auf dem Gebiet $[0,1)^2$ verwenden wir die bijektive Abbildung

$$\phi\left(\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}\right) := \begin{pmatrix} (x_1 + x_2) \mod 1 \\ x_1 \end{pmatrix}.$$

Diese ist auf $\{(x_1, x_2)^T \in [0,1)^2 \mid x_1 + x_2 = 1\}$ unstetig bezüglich der euklidischen Norm und bildet somit keinen Diffeomorphismus. [7] Auf dem Rest des Gebiets $[0,1)^2$ gilt $\det J_\phi = -1$ und der Prozess ist dort konservativ. Als Observable verwenden wir $o((x_1, x_2)^T) := x_1$. Mit dem Anfangswert $(0.15, 0.15)^T$ erzeugen wir die Zeitreihe T_{10^4}.
Sowohl der Autokorrelations- als auch der Mutual-Information-Schätzer zeigen, dass der Zeitparameter $\tau = 1$ bereits optimal ist. Um eine Dimensionsschätzung vorzunehmen,



(a) Boxcounting-Schätzer (b) Korrelationsdimensionsschätzer

Abb. 7.19: Schätzungen der Renyi-Dimensionen für 200 verschiedene ϵ_i anhand von T_{10^4}

wird die Zeitreihe in $\mathbb{R}^{10}$ eingebettet. Wie in Abbildung 7.19 zu sehen ist, liefert der Boxcounting-Schätzer kein Plateau. Der Korrelationsdimensionsschätzer ist allerdings in der Lage, die zugrundeliegende Dimension zu identifizieren. Wir nehmen nun an, nicht zu wissen, dass der ursprüngliche Prozess aus $\mathbb{R}^2$ stammt und betten die Zeitreihe in $\mathbb{R}^{2\cdot 2+1} = \mathbb{R}^5$ ein. Neben diesen Trainingsdaten dienen uns die auf die Zeitreihe T_{10^4} folgenden 10^4 Werte als Testdatensatz. Die Vorhersagegüte des Dünngitteralgorithmus wird nun bestimmt, indem mittels der eingebetteten Trainingsdaten die Funktion f_t^λ berechnet wird und die empirischen l_2-Fehler anhand einer Vorhersage auf den ebenfalls eingebetteten Testdaten kalkuliert werden.
Es ist nun ein geeigneter Parametersatz (t, λ) zur Vorhersage der Testdaten zu suchen. Am intuitivsten wäre es, die Parameter zu verwenden, welche auf den Trainingsdaten den niedrigsten Vorhersagefehler liefern. Diese Methode ist jedoch oftmals ungeeignet,

[7] Es sei auf den engen Zusammenhang zwischen der Struktur der Abbildung und dem flachen Torus $(\mathbb{R}/\mathbb{Z})^2$ hingewiesen, der in $\mathbb{R}^3$ einbettbar ist, siehe auch [doC92]

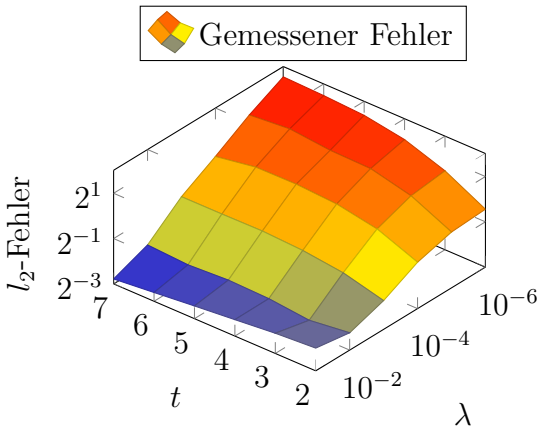
da es zum Phänomen der Überanpassung (*Overfitting*) kommen kann: Mit kleinem λ und großem t können hohe Erkennungsraten auf den Trainingsdaten erzielt werden. Solche Parameter eigenen sich aber meist nicht, um neue Testdaten vorherzusagen.
Aus diesem Grund wählen wir die Methode der *n-fachen Kreuzvalidierung*, siehe auch [Gar04]. Hierbei wird der Trainingsdatensatz in k gleichgroße Teile

$$D_c, c = 1, \ldots, k$$

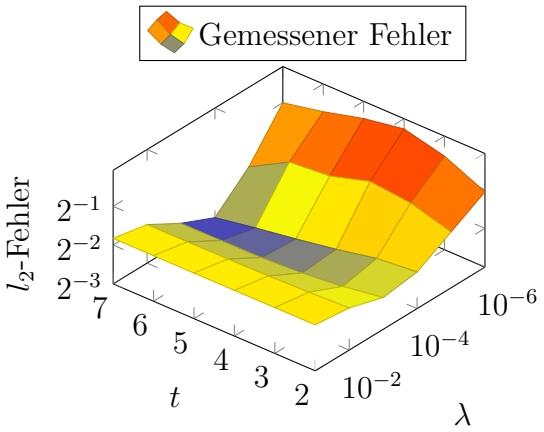
unterteilt. Nun berechnet man für jedes $c = 1, \ldots, k$ die Funktion ${}^c f_t^\lambda$ auf Grundlage der reduzierten Trainingsdaten

$$\bigcup_{i \neq c} D_i$$

mittels des Dünngitteralgorithmus und evaluiert den Fehler auf D_c. Das arithmetische Mittel der k Fehler liefert den Kreuzvalidierungsfehler für den Parametersatz (t, λ). Der Parametersatz, für welchen der Kreuzvalidierungsfehler minimal ist, wird nun für das ursprüngliche Problem verwendet.



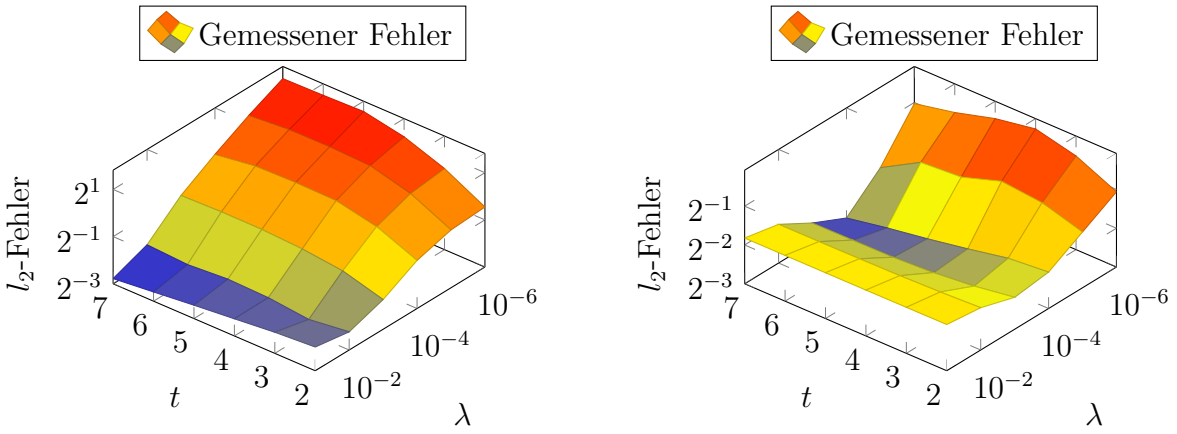
(a) l_2-Fehler, H^1-Regularisierung



(b) l_2-Fehler, H^1_{mix}-Regularisierung

Abb. 7.20: Arithmetisches Mittel der Fehler der 10-fachen Kreuzvalidierung für $t \in \{2, \ldots, 7\}$ und $\lambda \in \{10^{-1}, \ldots, 10^{-6}\}$

Die Ergebnisse der zehnfachen Kreuzvalidierung sind in Abbildung 7.20 zu sehen. Die Unterteilung der Trainingsdaten in die D_c wurde hierbei chronologisch vorgenommen. Die Kreuzvalidierung ergibt den resultierenden Parametersatz $l = 7, \lambda = 10^{-1}$ für die H^1-Regularisierung. Dieser führt auf den Trainingsdaten zu einem Kreuzvalidierungsfehler von 0.145. Der Fehler auf den Testdaten beträgt 0.146. Für die H^1_{mix}-Regularisierung findet man den besten Parametersatz $l = 7, \lambda = 10^{-4}$ mit einem mittleren l_2-Fehler von 0.134 auf den Trainingsdaten und einem Fehler von 0.133 auf den Testdaten. Die entsprechenden Fehler auf den Testdaten sind in Abbildung 7.21 dargestellt.



(a) l_2-Fehler, H^1-Regularisierung

(b) l_2-Fehler, H^1_{mix}-Regularisierung

Abb. 7.21: Fehler auf den Testdaten für $t \in \{2, \ldots, 7\}$ und $\lambda \in \{10^{-1}, \ldots, 10^{-6}\}$

Wie man sieht, findet man hier mittels der Kreuzvalidierung sowohl für die H^1- als auch für die H^1_{mix}-Regularisierung den Parametersatz (t, λ), der auch auf den Testdaten die besten Ergebnisse erzielt. Außerdem ist zu sehen, dass die Fehler für die Trainings- sowie die Testdaten nahezu gleich sind. Dies ist darauf zurückzuführen, dass die Trainingsdaten den Prozess bereits ausreichend gut beschreiben und dieser stationär ist. Für praxisrelevante Datensätze ist dies meist nicht der Fall.
Wir wollen nun das Abschneiden des dimensionsadaptiven Algorithmus beurteilen. Zu diesem Zweck sind in den Tabellen 7.23, 7.24 die l_2-Fehler auf den Testdaten nicht nur in Abhängigkeit von der globalen Fehlertoleranz, sondern auch von der Anzahl der während der Laufzeit berechneten Gitter sowie der Gesamtzahl ihrer Punkte dargestellt. Aus Gründen der Übersichtlichkeit werden dort nur wenige ausgewählte Resultate präsentiert, die das Verhalten der einzelnen Fehlerindikatoren deutlich machen sollen. Der Fehlerindikator $\epsilon^{\text{hier}}_{\mathbf{k}}$ wurde hierbei mit der L_2-Norm realisiert. Als Referenz dient Tabelle 7.22, in welcher die korresponiderenden Angaben für reguläre dünne Gitter zu finden sind.
Die präsentierten Ergebnisse sind mit der gewöhnlichen hierarchischen Basis berechnet worden. Die Wahl der Hierarchie mit konstanten Funktionen führt für vertretbaren Aufwand nicht zum erwünschten Konvergenzverhalten des Fehlers. Somit ist ein dimensionsadaptives Vorgehen im ANOVA-Sinn hier scheinbar nicht möglich. Für die H^1-Regularisierung ist der Algorithmus auch bei Verwendung der gewöhnlichen hierarchischen Basis nicht in der Lage bessere Ergebnisse als der reguläre Dünngitteralgorithmus zu liefern. Für die H^1_{mix}-Regularisierung erzielt die Vorgehensweise mit dem Fehlerindikator $|\epsilon^{l_2}_{\mathbf{k}}|$ die besten Ergebnisse und man erreicht einen zum regulären Verfahren auf Level 7 vergleichbaren Fehler mit nur halb so vielen Punkten. Der reguläre Algorithmus läuft dennoch in kürzerer Zeit, da die Auswertung des Fehlerindikators $|\epsilon^{l_2}_{\mathbf{k}}|$ den Aufwand dominiert. Ein ähnliches Fehlerverhalten ist auch für andere λ zu beobachten.

t	Gitter	Punkte	l_2-Fehler (H^1)	l_2-Fehler (H^1_{mix})
2	21	1,392	0.251	0.240
3	56	5,592	0.232	0.225
4	126	19,482	0.207	0.195
5	251	61,813	0.185	0.171
6	456	183,698	0.164	0.152
7	771	520,333	0.146	0.133

Tab. 7.22: l_2-Fehler auf den Testdaten für reguläre dünne Gitter – bei H^1-Regularisierung für $\lambda = 0.1$, bei H^1_{mix}-Regularisierung für $\lambda = 0.0001$

Toleranz	Gitter	Punkte	l_2-Fehler
$1.0{\cdot}10^{-3}$	88	13,828	0.230
$5.0{\cdot}10^{-4}$	198	128,348	0.188
$2.5{\cdot}10^{-4}$	751	818,768	0.168
$1.0{\cdot}10^{-4}$	1,504	3,721,378	0.135

(a) $\epsilon^{\text{hier}}_{\mathbf{k}}$

Toleranz	Gitter	Punkte	l_2-Fehler
$5.0{\cdot}10^{-4}$	24	4,016	0.251
$1.0{\cdot}10^{-4}$	104	46,676	0.216
$5.0{\cdot}10^{-5}$	139	93,668	0.205
$1.0{\cdot}10^{-5}$	260	669,992	0.191

(b) $\epsilon^{l_2}_{\mathbf{k}}$

Toleranz	Gitter	Punkte	l_2-Fehler
$1.0{\cdot}10^{-4}$	109	36,258	0.216
$5.0{\cdot}10^{-5}$	170	95,246	0.198
$2.5{\cdot}10^{-5}$	297	450,464	0.176
$1.0{\cdot}10^{-5}$	639	1,170,504	0.138

(c) $|\epsilon^{l_2}_{\mathbf{k}}|$

Toleranz	Gitter	Punkte	l_2-Fehler
$5.0{\cdot}10^{-3}$	15	1,136	0.260
$1.0{\cdot}10^{-3}$	185	62,802	0.191
$5.0{\cdot}10^{-4}$	503	330,030	0.149
$2.5{\cdot}10^{-4}$	1,381	2,740,120	0.118

(d) $\epsilon^{f}_{\mathbf{k}}$

Tab. 7.23: l_2-Fehler auf den Testdaten in Abhängigkeit von den Fehlerindikatoren aus Unterabschnitt 6.3.2 für H^1-Regularisierung mit $\lambda = 10^{-1}$

Auch wenn die verwendeten Fehlerindikatoren für die Hierarchie mit konstanten Funktionen keine guten Ergebnisse liefern, ist für den Indikator $\epsilon^f_{\mathbf{k}}$ ein interessantes Verhalten zu beobachten: Für $\lambda = 10^{-3}$ im H^1-Fall und $\lambda = 10^{-6}$ im H^1_{mix}-Fall oszilliert der l_2-Fehler auf den Trainingsdaten während des iterativen Berechnens der Indexmenge $\mathbf{I}$ sehr stark. Im Fall eines ausreichend großen Samplings und eines stationären Prozesses ist anzunehmen, dass sich der Fehler auf der Testdatenmenge ähnlich verhält. Dies können wir nutzen, um einen erweiterten Fehlerindikator zu konstruieren, indem eine zusätzliche Grenze $B \in \mathbb{R}^{++}$ eingeführt wird. Der dimensionsadaptive Algorithmus berechnet $\mathbf{I}$ nach dem bereits bekannten Schema, bricht allerdings ab, sobald der l_2-Fehler auf den Trainingsdaten die Grenze B unterschritten hat. Diese Erweiterung lässt sich problemlos auf

Toleranz	Gitter	Punkte	l_2-Fehler
$5.0{\cdot}10^{-4}$	155	40,072	0.216
$1.0{\cdot}10^{-4}$	372	177,458	0.189
$5.0{\cdot}10^{-5}$	499	312,534	0.181
$1.0{\cdot}10^{-5}$	870	984,660	0.149

(a) $\epsilon_{\mathbf{k}}^{\text{hier}}$

Toleranz	Gitter	Punkte	l_2-Fehler
$1.0{\cdot}10^{-4}$	20	18,408	0.174
$5.0{\cdot}10^{-5}$	22	36,368	0.174
$1.0{\cdot}10^{-5}$	25	137,800	0.174
$5.0{\cdot}10^{-6}$	27	275,056	0.174

(b) $\epsilon_{\mathbf{k}}^{l_2}$

Toleranz	Gitter	Punkte	l_2-Fehler
$1.0{\cdot}10^{-3}$	136	28,614	0.157
$5.0{\cdot}10^{-4}$	228	62,673	0.143
$2.5{\cdot}10^{-4}$	382	160,721	0.138
$1.0{\cdot}10^{-4}$	478	262,195	0.133

(c) $|\epsilon_{\mathbf{k}}^{l_2}|$

Toleranz	Gitter	Punkte	l_2-Fehler
$5.0{\cdot}10^{-2}$	13	1,032	0.190
$1.0{\cdot}10^{-2}$	171	40,785	0.149
$5.0{\cdot}10^{-3}$	446	201,943	0.138
$2.5{\cdot}10^{-3}$	797	597,135	0.132

(d) $\epsilon_{\mathbf{k}}^{f}$

Tab. 7.24: l_2-Fehler auf den Testdaten in Abhängigkeit von den Fehlerindikatoren aus Unterabschnitt 6.3.2 für H^1_{mix}-Regularisierung mit $\lambda = 10^{-4}$

die Indikatoren $\epsilon_{\mathbf{k}}^{l_2}$, $|\epsilon_{\mathbf{k}}^{l_2}|$ und $\epsilon_{\mathbf{k}}^{f}$ anwenden, da bei deren Verwendung die Berechnung des l_2-Fehlers nach jedem Iterationsschritt mit höchstens konstantem Mehraufwand erfolgen kann.
Mithilfe des so modifizierten Fehlerschätzers $\hat{\epsilon}_{\mathbf{k}}^{f}$ lassen sich mittels der Hierarchie mit konstanten Funktionen gute Ergebnisse erzielen. Somit liefert auch ein im ANOVA-Sinn "echter" dimensionsadaptiver Algorithmus hier gute Ergebnisse. Eine Auswahl der Resultate ist in Tabelle 7.25 dargstellt.

B	Gitter	Punkte	l_2-Fehler
0.12	69	4,981	0.120
0.11	183	55,567	0.111
0.10	214	95,132	0.099
0.09	256	291,680	0.092

(a) H^1-Regularisierung ($\lambda = 10^{-3}$)

B	Gitter	Punkte	l_2-Fehler
0.12	82	140,941	0.130
0.11	105	166,939	0.125
0.10	114	206,924	0.115
0.09	512	3,186,652	0.108

(b) H^1_{mix}-Regularisierung ($\lambda = 10^{-6}$)

Tab. 7.25: l_2-Fehler auf den Testdaten für den Fehlerindikator $\hat{\epsilon}_{\mathbf{k}}^{f}$ mit Toleranz 10^{-3} für die Hierarchie mit konstanten Funktionen

Insbesondere das Resultat für $B = 0.12$ bei einer H^1-Regularisierung ist bemerkenswert, da das beste Ergebnis für reguläre dünne Gitter deutlich übertroffen wird und die Zahl aller betrachteten Gitterpunkte niedriger ist als die eines regulären Gitters für $t = 3$.

Des Weiteren ist anzumerken, dass die Laufzeiten der Fälle $B = 0.12$ bei H^1- und $B = 0.12, 0.11, 0.10$ bei H^1_{mix}-Regularisierung unter gleichen Voraussetzungen um mehr als den Faktor 200 mal kleiner waren als die Laufzeit zur Berechnung der Lösung auf einem regulären dünnen Gitter des Levels 7.
Dass die Struktur des Problems korrekt erkannt wird, sieht man vor allem an den Gittern, welche mit einem Nicht-Null-Koeffizienten in die Kombinationstechnik eingehen. Für $B = 0.12$ gehen sowohl im H^1 als auch im H^1_{mix}-Fall ausschließlich Gitter ein, welche zu Basisfunktionen korrespondieren, die in maximal zwei Koordinatenrichtung nicht konstant sind. Somit wird die Superpositionsdimension 1 der zu approximierenden Funktion nur leicht überschätzt. Welche Dimensionen im Einzelnen verfeinert werden, ist nahezu belanglos, da durch die Struktur der Delay-Einbettung jedes Paar aufeinanderfolgender Koordinatenrichtungen genügend Informationen enthält.
Auch der ortsadaptive Algorithmus ist in der Lage bessere Resultate als der gewöhnliche Dünngitteralgorithmus zu erzielen. Hier wurde ebenfalls die L_2-Norm zur Auswertung des Fehlerindikators $\epsilon^{\text{hier}}_{\mathbf{x_{l,i}}}$ verwendet.
Um eine substantielle Reduktion der Freiheitsgrade vorzunehmen, ist es ratsam, auf dem Rand des Gebiets $[0,1]^5$ keine Gitterpunkte zuzulassen. Das Startgitter Ω^s_2 besteht dann aus nur $2d + 1 = 11$ statt $3^d + 2d \cdot 3^{d-1} = 1053$ Punkten. Ist der Wert der zu approximierenden Funktion auf dem gesamten Rand gleich Null, so verschlechtert diese Maßnahme die Approximation nicht. In [BPZ08] wird gezeigt, dass eine ortsadaptive Variante ohne Randdiskretisierung auch für einen Nicht-Null-Rand sinnvoll ist und gute Ergebnisse erzielt, sofern die vorliegenden Datenpunkte so skaliert werden, dass kein Datenpunkt auf den Rand des Gebiets fällt. Aus diesem Grund haben wir die Zeitreihe hier auf $[0.1, 0.9]$ statt $[0, 1]$ skaliert und dann eine ortsadaptive Rechnung ohne Randdiskretisierung durchgeführt. Eine Auswahl der Ergbenisse findet sich in Tabelle 7.26.

L	Punkte	l_2-Fehler
5	1,872	0.139
6	6,949	0.114
7	20,996	0.096
8	48,446	0.087
9	89,220	0.080
10	142,450	0.075

(a) H^1-Regularisierung (Toleranz 10^{-3}, $\lambda = 10^{-3}$)

L	Punkte	l_2-Fehler
5	1,883	0.145
6	6,599	0.126
7	14,069	0.114
8	22,325	0.110
9	22,325	0.110
10	22,325	0.110

(b) H^1_{mix}-Regularisierung (Toleranz 10^{-5}, $\lambda = 10^{-7}$)

L	Punkte	l_2-Fehler
5	1,904	0.145
6	7,387	0.126
7	26,027	0.113
8	82,123	0.105
9	208,834	0.100
10	406,864	0.096

(c) H^1_{mix}-Regularisierung (Toleranz 10^{-5}, $\lambda = 10^{-8}$)

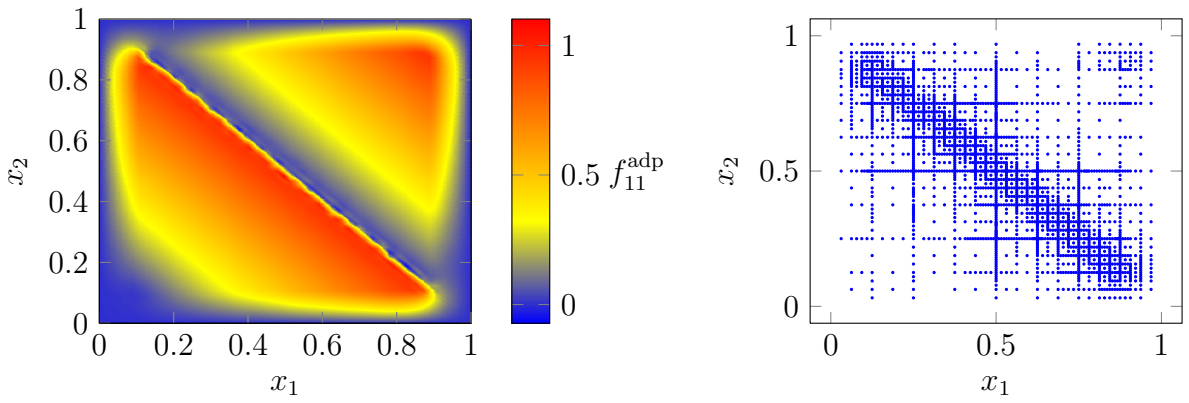
Tab. 7.26: l_2-Fehler auf den Testdaten für den ortsadaptiven Algorithmus mit Startgitter Ω^s_2 und maximalem Level L

Um den Aufwand mit dem des dimensionsadaptiven Algorithmus vergleichen zu können,

wird die Summe

$$\sum_{j=2}^{L} |\Omega_j^{\text{adp}}|$$

der Punkte aller Gitter bis zum Level L angegeben. Wie zu sehen ist, ist es möglich den Fehler, welcher mit einem regulären Gitter des Levels 7 erzielt werden kann, bereits mit einem Bruchteil des Aufwands zu erreichen. Sogar die Approximationsgüte des dimensionsadaptiven Algorithmus wird übertroffen. Im Fall der H^1_{mix}-Regularisierung für $\lambda = 10^{-7}$ wird ab Level 8 nicht weiter verfeinert, da der Fehlerindikator für keinen Punkt die Toleranzgrenze überschreitet.
In Abbildung 7.27 ist anhand einer zweidimensionalen Einbettung zu sehen, dass der ortsadaptive Algorithmus in der Lage ist, die Struktur der zu approximierenden Funktion zu erkennen und ein angepasstes Gitter zu generieren.



(a) Errechnete Approximation f_{11}^{adp} (b) Adaptives Gitter Ω_{11}^{adp}

Abb. 7.27: H^1-regularisierte Approximation auf dem ortsadaptiven Gitter Ω_{11}^{adp} – Zur Darstellung der Funktion wurde diese auf das reguläre Gitter Ω_6 interpoliert

7.4 Vergleich der Regularisierungen

Dass es Fälle gibt, in denen eine H^1_{mix}-Regularisierung bedeutend bessere Ergebnisse erzielen kann als eine H^1-Regularisierung, soll in diesem Experiment gezeigt werden. Hierzu verwenden wir den zweidimensionalen zeitdiskreten Prozess

$$\phi\left(\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}\right) := \begin{pmatrix} \frac{1}{k}\cos(kx_1x_2) \\ \log(|x_1|) \end{pmatrix}$$

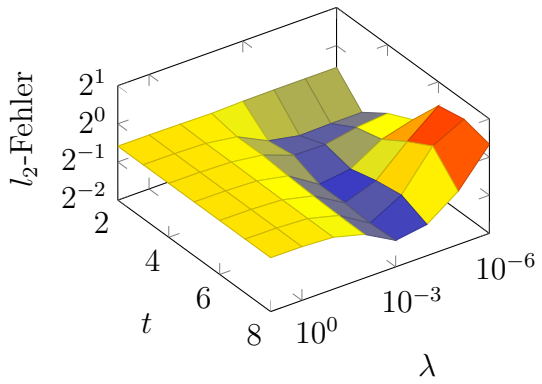
auf $[-1,1] \times \mathbb{R}$ mit $k \in \mathbb{R}$ und die Observable $o((x_1,x_2)^T) := k \cdot x_1$.[8] Es gilt

$$|\det \mathbf{J}_\phi| = |-\sin(kx_1x_2)| \leq 1.$$

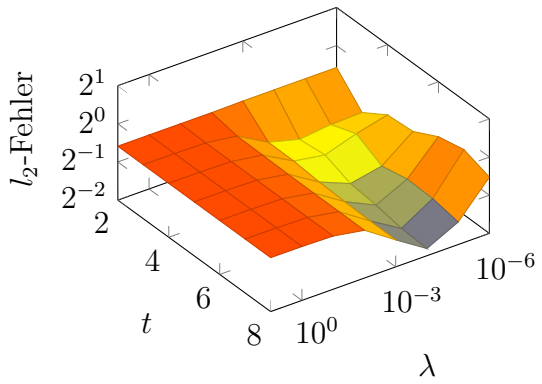
Neben der Beschränktheit der ersten Ableitung ist zu sehen, dass

$$\left| \frac{\partial^2}{\partial x_1 \partial x_2} \left(\frac{1}{k} \cos(kx_1x_2) \right) \right| \overset{k\to\infty}{\to} \infty$$

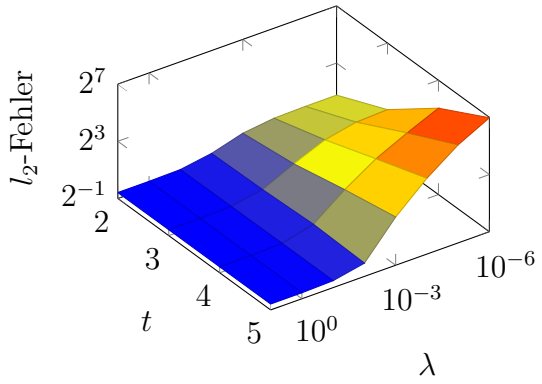
gilt. Je größer k wird, umso mehr trägt der gemischte Term zur Regularisierung bei.



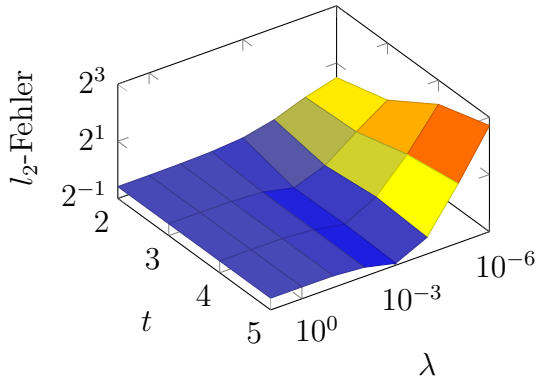
(a) l_2-Fehler, H^1-Regularisierung, $d = 2$



(b) l_2-Fehler, H^1_{mix}-Regularisierung, $d = 2$



(c) l_2-Fehler, H^1-Regularisierung, $d = 5$



(d) l_2-Fehler, H^1_{mix}-Regularisierung, $d = 5$

Abb. 7.28: l_2-Fehler auf 250 Testdaten für $k = 10^5$ für die Einbettungsdimensionen $d = 2, 5$

Da eine große Trainingsdatenmenge für stationäre Prozesse auch ohne Regularisierung zu sehr guten Resultaten auf den Testdaten führt, beschränken wir uns in diesem Beispiel auf die Zeitreihe T_{250} und erstellen ein Modell, welches auf den darauffolgenden 250

[8] Dieser Prozess ist offensichtlich nicht bijektiv.

Werten getestet wird. Als Anfangswert wurde $\mathbf{z_0} = (1,1)^T$ verwendet. In Abbildung 7.28 sind die Ergebnisse für $k = 10^5$ dargestellt.
Es ist zu sehen, dass der Fehler für $\lambda \geq 10^{-1}$ auf allen Leveln nahezu identisch ist. Dies ist darauf zurückzuführen, dass die resultierenden Funktionen sehr glatt sind. Für $d = 2$ ist die Struktur des Fehlerverhaltens für beide Regularisierungen ähnlich, auch wenn der minimale Fehler eines H^1-regularisierten Verfahrens mit 0.405 deutlich größer ist als der eines H^1_{mix}-regularisierten mit 0.285. Für $d = 5$ wird die beste Approximation mit H^1-Regularisierung für $\lambda = 10$ erreicht und ist auf allen Leveln 0.666. Eine bessere Approximation für ein kleineres λ ist hier nicht möglich, da die H^1-Regularisierung dann zu schwach ist und es auf den Trainingsdaten zu einer Überanpassung kommt. Der H^1_{mix}-regularisierte Algorithmus liefert den kleinsten Fehler 0.525 hingegen für $\lambda = 10^{-3}$ und Level 5. Die Fehler für $\lambda = 10^{-2}, 10^{-3}$ auf den Leveln $3, 4$ und 5 sind hier kleiner als der Fehler 0.667 für $\lambda \geq 10^{-1}$ und die H^1_{mix}-Regularisierung führt zu einem Modell, welches auf allgemeinen Testdaten besser abschneidet als das H^1-regularisierte.

7.5 Vergleiche mit anderen Verfahren

7.5.1 Vorhersage von Wechselkursdaten

In [GGG09] wird die Methode der Dünngitterapproximation zur Vorhersage von Wechselkursdaten verwendet. Wir analysieren den dort beschriebenen *Olsen*-Datensatz und beschränken uns auf den €/$-Wechselkurs. Es liegen 701280 Kursraten in äquidistanten Abständen von drei Minuten zwischen August 2001 und August 2005 vor, wobei insgesamt 168952 der Raten als "ungültig" markiert und somit nicht zu verwenden sind. [9]
Für eine genauere Auseinandersetzung mit der Struktur der Daten und der Behandlung ungültiger Kursraten sei auf [GGG09] verwiesen.
Um diesen nichtstationären Datensatz besser analysieren zu können, verwenden wir nicht die vorliegenden Rohdaten, sondern die Differenzen zwischen zwei aufeinanderfolgenden Kursraten. Die Delay-Schätzer signalisieren beide, dass $\tau = 1$ bereits die bestmögliche Wahl des Zeitparameters ist. Die Dimensionsschätzer liefern keine brauchbaren Resultate: Die PCA und lokale PCA zeigen im $\mathbb{R}^{100}$ keinen signifikanten Abfall der Eigenwerte der Kovarianzmatrix und die Renyi-Dimensionsschätzer liefern im $\mathbb{R}^{10}$ kein eindeutiges Plateau. [10]
Dennoch ist in [GGG09] zu sehen, dass eine Einbettung des Prozesses und eine anschließende Vorhersage mit der Dünngittermethode profitable Ergebnisse erzeugt. Wir wollen dies nun ebenfalls experimentell nachweisen, gehen allerdings anders vor.

[9] Die Kursrate zu einem bestimmten *Tick* stellt hierbei das arithmetische Mittel aus *Bid*- und *Ask*-Preis dar.

[10] Die Delay-Schätzer operieren für eine Schätzung zum Zeitparameter τ ausschließlich auf Datenpunkten zwischen denen kein ungültiger Tick lag. Analoges gilt für die Einbettung der Daten zur Verwendung in den Dimensionsschätzern.

Zum Zeitpunkt t betten wir die normierten Differenzen

$$f_{t,k} := \frac{f(t) - f(t - k\hat{\tau})}{f(t - k\hat{\tau})}$$

für $k = 1, \ldots, 15$ ein, wobei $f(s)$ den Kurs zum Zeitpunkt s bezeichnet und $\hat{\tau} = 3\,\text{min}$ ist. Unser Ziel ist es nun – analog zu [GGG09] – eine Prognose über den 45 Minuten später vorliegenden Kursstand zu treffen und somit eine Vorhersage des Wertes

$$f_{t+15\cdot\hat{\tau},15} = \frac{f(t + 15 \cdot \hat{\tau}) - f(t)}{f(t)}.$$

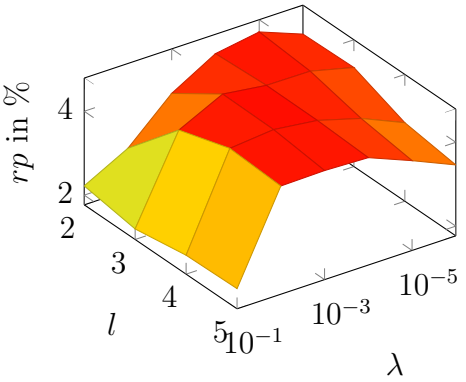
zu errechnen. Wir analysieren dieses 15-dimensionale Problem nun mit der dimensionsadaptiven Kombinationstechnik. Als Trainingsdaten dienen uns die Daten vom 1. August 2001 bis zum 8. März 2005. Dies entspricht 90% des gesamten Datensatzes. Die restlichen 10% verwenden wir als Testdatensatz zur Evaluierung unseres Modells. Die Komplexität des Problems ist hier zu groß um alle relevanten Richtungen so weit zu verfeinern, dass gute Resultate erzielt werden. Dennoch zeigt die adaptiv konstruierte Indexmenge $\mathbf{I}$ bereits nach wenigen Iterationen eine deutliche Struktur. Für ein H^1_{mix}-regularisiertes Verfahren mit $\lambda = 10^{-4}$ und einer Toleranzgrenze von 10^{-3} bei Verwendung des Fehlerindikators $\epsilon^f_{\mathbf{k}}$ betrachten wir die Indexmenge $\mathbf{I}$ nach dem Einfügen der ersten 150 Indizes. Mehr als 90% der eingfügten Gitter sind in den zu $k = 4, 5, 11$ und 12 korrespondierenden Richtungen verfeinert. Andere Richtungen treten wesentlich seltener auf. Dieses Resultat motiviert die Betrachtung des reduzierten vierdimensionalen Problems. Als Eingabepunkte dienen uns nun die Vektoren $(f_{t,4}, f_{t,5}, f_{t,11}, f_{t,12})^T$. Zur Bestimmung geeigneter Parameter λ und Level l führen wir eine dreifache Kreuzvalidierung auf den Trainingsdaten für $\lambda = 10^{-1}, \ldots, 10^{-6}$ und $l = 2, \ldots, 5$ durch. Wir suchen den Parametersatz, welcher den höchsten realisierten Profit (rp) erzielt, wobei

$$rp := \frac{\sum_t \text{sign}(g^c_l)(\mathbf{x}_t) \cdot (f(t + 15 \cdot \hat{\tau}) - f(t))}{\sum_t f(t + 15 \cdot \hat{\tau}) - f(t)}$$

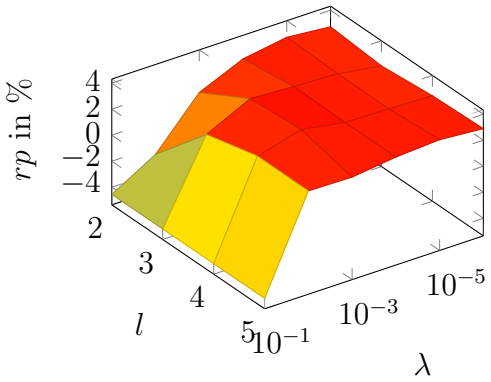
gilt. g^c_l ist hierbei die errechnete Vorhersagefunktion und $\mathbf{x}_t$ der eingebettete Vektor zum Zeitpunkt t. Die Resultate der Kreuzvalidierung sowie die Ergebnisse auf den Testdaten sind in Abbildung 7.29 zu sehen. Der maximale rp von 4.54% auf den Trainingsdaten wird für $\lambda = 10^{-3}$ und $l = 3$ erzielt. Für diesen Parametersatz ergibt sich ein rp von 3.23% auf den Testdaten. Die Erkennungsrate

$$\frac{|\{t \mid g^c_l(\mathbf{x}_t) \cdot f_{t+15\cdot\hat{\tau},15} > 0\}|}{|\{t \mid g^c_l(\mathbf{x}_t) \cdot f_{t+15\cdot\hat{\tau},15} \neq 0\}|}$$

liegt bei 52.03%. Verwendet man diese Parameter und geht lediglich zu den Zeitpunkten t_{border} eine Position ein, an welchen $|g^c_l(\mathbf{x}_t)|$ größer als 10^{-4} ist, so entspricht dies 960 von 50653 möglichen Aktionen auf den Testdaten und führt zu einem rp von 20.81%



(a) Kreuzvalidierung (gemittelter rp auf den Trainingsdaten)



(b) rp auf den Testdaten

Abb. 7.29: rp für das vierdimensionale Wechselkurs-Problem (H^1_{mix}-Regularisierung)

sowie einer Erkennungsrate von 57.93%. Diese Ergebnisse sind mit denen aus [GGG09] vergleichbar, basieren allerdings auf anderen Lerndaten.[11]

7.5.2 Datensatz D des "Santa Fe"-Wettbewerbs

Dieser Datensatz besteht aus 100000 Zeitreihendaten, die im Lernprozess verwendet werden sollen sowie weiteren 500 Testdaten, die zur Vorhersage dienen. Der Datensatz entsteht durch das Anwenden eines Runge-Kutta-Verfahrens vierter Ordnung zur Lösung von

$$\frac{\partial^2}{\partial t^2}\mathbf{x}(t) + \gamma\frac{\partial}{\partial t}\mathbf{x}(t) + \nabla V(\mathbf{x}(t)) = c \cdot \sin(\omega t)$$

mit $\mathbf{x} := (x_1, x_2, x_3, x_4)^T$ und

$$V(\mathbf{x}) := a_4 \cdot ||\mathbf{x}||_2 - a_2(x_1x_2)^2 - a_1x_1.$$

Als Observable dient

$$\sqrt{(x_1^2 + 0.3)^2 + (x_2^2 + 0.3)^2 + x_3^2 + x_4^2}.$$

Genaue Angaben zur Wahl der Parameter sowie allgemeine Informationen zum "Santa Fe"-Wettbewerb sind in [WG92] zu finden.
Wir verwenden im Folgenden lediglich die ersten 25 Werte des gesamten Testdatensatzes. Dies ermöglicht es unsere Resultate mit den Ergebnissen in [MSR+00] zu vergleichen. Als Fehlermaß dient uns der l_2-Datenfehler auf den Testdaten (RMSE).

[11] Ein ortsadaptiver Ansatz konnte in unseren Experimenten keine substantiell besseren Ergebnisse erreichen.

Um den nichtstationären Anteil nicht zu stark in das Problem eingehen zu lassen, verwenden wir nur die letzten 2000 Trainingsdaten und teilen diese nun in die ersten 1975 (*Trainingsmenge*) und die letzten 25 (*Evaluierungsmenge*) Werte auf. Beide Delay-Schätzer liefern das Resultat $\tau = 1$ als optimalen Zeitparameter. Die Dimensionsschätzer sind (auch auf dem gesamten Datensatz) nicht in der Lage eine stabile Schätzung zu liefern. Um die 25 aufeinanderfolgenden Testdaten vorhersagen zu können, wählen wir folgende Strategie:
Für jedes $k \in \{1, \ldots, 25\}$ wird mittels des H^1_{mix}-regularisierten Dünngitteralgorithmus auf der Trainingsmenge für alle $\lambda \in \{0.5, 0.1, 0.05, 0.01, 0.005, 0.001, 0.0005, 0.0001\}$ und Level $l \in \{2, 3, 4, 5, 6, 7\}$ eine Funktion berechnet, die zum aktuellen Zeitpunkt t eine Approximation des Zeitreihenwerts zum Zeitpunkt $t + k$ liefert. Auf der Evaluierungsmenge wird der l_2-Fehler dieser Approximationen bestimmt. Zur Berechnung der endgültigen Approximation wählen wir nun die Parameter $\hat{k}, \hat{\lambda}, \hat{l}$, für welche der l_2-Fehler auf der Evaluierungsmenge am geringsten ist und erstellen mit Hilfe der Trainings- und Evaluierungsmenge die Approximation $\hat{g}_{\hat{k}}$, welche verwendet wird, um die Zeitreihe auf den Testdaten vorherzusagen. Für alle $\hat{k} < 25$ tritt der Fall ein, dass wir zur Vorhersage auf den Testdaten eingebettete Punkte $\mathbf{x}_i$ benötigen, die nicht anhand der Trainings- oder Evaluierungsmenge berechnet werden können. In diesem Fall approximieren wir die fehlende Koordinate durch die Vorhersage $\hat{g}_{\hat{k}}(\mathbf{x}_{i-\hat{k}})$.[12]
Wir verfolgen die beschriebene Strategie für Einbettungen in $\mathbb{R}^d$ mit $d = 1, \ldots, 4$. Der geringste l_2-Fehler (0.0686) auf der Evaluierungsmenge resultiert für die Parameter $d = 3$, $k = 2$, $\lambda = 0.01$ und $l = 2$. Mit diesen Parametern erreichen wir einen l_2-Fehler von 0.0639 auf den Testdaten und erzielen somit vergleichbare Resultate zu [MSR+00], wo mittels einer Support-Vektor-Maschine der geringste l_2-Fehler ohne Vorverarbeitung der Daten – also analog zu unserer Vorgehensweise – ebenfalls 0.0639 beträgt. Unsere Vorgehensweise schlägt somit den dort ebenfalls angeführten Ansatz mit radialen Basisfunktionen, der einen l_2-Fehler von 0.0677 erzielt. In Abbildung 7.30 sind die letzten 75 Punkte der Trainingsdaten sowie die 25 von uns vorhergesagten und die 25 echten Testdaten dargestellt.

Mittels eines ortsadaptiven Ansatz erhalten wir für $d = 4$, $\lambda = 10^{-4}$, $k = 1$, einer Toleranzgrenze von 10^{-9} und einem maximalen Level $l_{\max} = 10$ ein noch besseren l_2-Fehler von 0.0608. Die Anpassung der Parameter fand hier allerdings auf den Testdaten statt.

7.5.3 Reduzierter Datensatz der "ANN and CI Competition 2006/2007"

Der reduzierte Datensatz der "Artificial Neural Network and Computational Intelligence Forecasting Competition 2006/2007" [ANN] besteht aus elf reellwertigen Zeitreihen aus

[12] Ist $\mathbf{x}_{i-\hat{k}}$ ebenfalls noch nicht vorhanden, verfahren wir analog zur Berechnung dieses Punktes und iterieren den Prozess, bis ausschließlich bekannte Koordinaten verwendet werden.

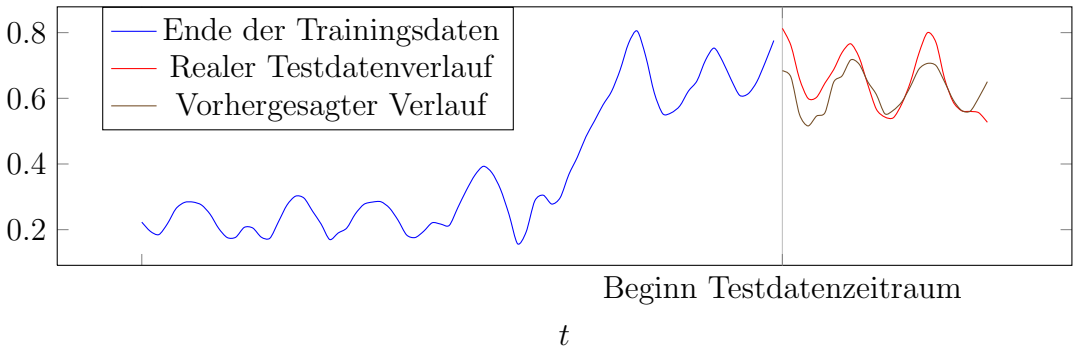


Abb. 7.30: Letzte 75 Punkte der Trainingsdaten und vorhergesagte sowie echte Testdaten des Datensatzes D des "Santa Fe"-Wettbewerbs

dem Wirtschaftsbereich. Im Rahmen des Wettbewerbs wurden 52 verschiedene Ansätze vorgeschlagen, die neuronale Netze verwenden, um die Zeitreihen vorherzusagen. Jede Zeitreihe besteht aus 126 Trainings- und 18 Testdaten. In den angegebenen Wettbewerbsbedingungen wird gefordert, dass derselbe Algorithmus auf jede der Zeitreihen angewendet wird und dass die Vorhersagen ohne Einfluss von außen stattfinden. Auf den Testdaten wird der *SMAPE* (*Symmetric Mean Absolut Percent Error*)

$$\frac{1}{18}\sum_{i=1}^{18}\frac{2\cdot|s_i-\hat{s}_i|}{(|s_i|+|\hat{s}_i|)}$$

errechnet, wobei s_i den echten Testdatenwert und $\hat{s}_i$ die errechnete Prognose darstellt. Das arithmetische Mittel der Fehler für die elf Zeitreihen stellt schließlich das endgültige Fehlermaß dar.
In unseren Experimenten hat sich eine Dimensionsbestimmung durch die PCA im $\mathbb{R}^5$ als unnütz herausgestellt, da oftmals zwar ein gewisser Abfall der Eigenwerte erkennbar war, die geschätzten Dimensionen jedoch meistens selbst für 80% Varianzerhaltung größer als 3 waren, was aufgrund der wenigen Trainingsdaten zu einer schlechten Vorhersagefunktion führt. Die Renyi-Schätzer sind hier ohnehin nicht verwendbar, da die Anzahl der Daten zu gering ist. Aus diesem Grund verfolgen wir dieselbe Strategie wie im "Santa Fe"-Beispiel und verwenden die letzten 18 Werte der Trainingsdaten als Evaluierungsmenge. Für jede der Zeitreihen errechnen wir den SMAPE auf der Evaluierungsmenge für sämtliche Kombinationen aus $\lambda \in \{0.5, 0.1, 0.05, 0.01, 0.005, 0.001, 0.0005, 0.0001\}$, $l \in \{2,3,4,5,6,7\}$, $k \in \{1,\ldots,18\}$ und $d \in \{1,2,3\}$. Der beste Parametersatz wird zur Vorhersage der Testdaten verwendet. Mit dieser Strategie erhalten wir einen mittleren SMAPE von 16.14%. Somit erzielen wir ein besseres Resultat als 44 der 52 Teilnehmer und als die meisten statistischen Methoden, welche als Vergleichsverfahren am Wettbewerb teilnahmen.

8 Schlussbemerkungen

8.1 Zusammenfassung

Zunächst haben wir den Begriff der Zeitreihe als Beobachtung eines Prozesses durch eine Observable definiert. Dieses allgemeine Konzept wurde am Beispiel einer aus dem Lorenz-System resultierenden Zeitreihe erklärt. Daraufhin wurden dissipative Prozesse und ihre Attraktoren formal definiert und die betreffenden Konzepte erläutert. Es wurde darauf hingewiesen, dass die meisten praxisrelevanten Prozesse dissipativ sind und sich auf einen Attraktor zubewegen. Anhand der Henon-Abbildung und des Lorenz-Systems wurden die Begriffe veranschaulicht.
Wir haben Takens' Theorem zur Rekonstruktion von Trajektorien zeitdiskreter dynamischer Prozesse in einem reellen Vektorraum kennengelernt. Es wurde kurz erwähnt, dass es möglich ist, die Voraussetzungen in Takens' Theorem abzuschwächen und dass es insbesondere ausreicht, den Dimensionsbegriff der Boxcounting-Dimension zugrundezulegen. Die Voraussetzungen an den Diffeomorphismus ϕ, welcher die Trajektorie bestimmt, sowie die Observable o wurden genauer erläutert. Eine Variante des Theorems für zeitkontinuierliche Prozesse sowie die Möglichkeit, die kennengelernte Delay-Einbettung durch eine Einbettung von Ableitungen zu ersetzen, wurden ebenfalls erklärt. Es wurde kurz erwähnt, dass eine Beziehung zu Whitneys Einbettungssatz, dem Theorem von Menger-Nöbeling und Kolmogorovs Superpositionstheorem besteht.
Bei der Anwendung der Delay-Einbettung in der Praxis ergeben sich mehrere Probleme. Insbesondere die Bestimmung einer hinreichend großen Einbettungsdimension und die Wahl des Delay-Parameters wurden detailliert diskutiert. Der Begriff der Renyi-Dimension wurde eingeführt und wir haben den Zusammenhang zwischen der Boxcounting- und der Hausdorff-Dimension skizziert. Es wurden Schätzer für die Boxcounting- sowie die Korrelationsdimension eingeführt und auf ihre Eignung in der Praxis untersucht. Für beide Dimensionsbegriffe haben wir schnelle Schätzer implementiert, deren Komplexität der eines naiven Ansatzes überlegen ist. M ittels der Hauptachsenzerlegung wurde ein alternativer Schätzer eingeführt. Wir haben den Concentration of Measure-Effekt erläutert und schließlich den direkten Zusammenhang zwischen Bregman-Divergenzen und exponentiellen Familien von Wahrscheinlichkeitsverteilungen kennengelernt. Zum Schätzen des Zeitparameters wurden die lineare Autokorrelationsmethode sowie der über die Shannon-Entropie motivierte Mutual Information-Ansatz erläutert.
Um eine Zeitreihe anhand eines eingebetteten Prozesses vorherzusagen, wurde ein Lernfunktional eingeführt, welches über die empirische Verteilung der vorliegenden Daten

sowie ein Glattheitsfunktional motiviert wurde. Der Zusammenhang zwischen der Regularisierung und Hilberträumen mit reproduzierendem Kern wurde kurz skizziert und begründete die Verwendung der H^1_{mix}-Norm im Lernfunktional. Die Minimierung des Funktionals führte zu einem unendlichdimensionalen linearen Gleichungssystem. Um das Problem der in der Datenmenge schnell ansteigenden Komplexität bei einem Kernansatz zu umgehen, wurde eine Diskretisierung mittels stückweise linearen finiten Elementen gewählt. Damit der Fluch der Dimension lediglich in abgeschwächeter Form auftritt, wurde eine Dünngitterdiskretisierung mittels der hierarchischen Faber-Basis vorgestellt. Wir haben die Kombinationstechnik kennengelernt, welche es ermöglicht die Dünngitterlösung durch das Lösen einfacher Teilprobleme auf vollen Gittern zu erhalten. Des Weiteren wurde über das Konzept der ANOVA-Zerlegung eine dimensionsadaptive Variante motiviert. Zuletzt wurde erwähnt, dass die Lokalität des Attraktors dazu führt, dass ein ortsadaptiver Ansatz sinnvoll sein kann, was sich in unseren Experimenten bestätigt hat. Wir haben anhand zweier Benchmark-Datensätze gesehen, dass die hier vorgestellten Algorithmen auch in der Praxis eine gute Alternative zu anderen Verfahren darstellen.

8.2 Ausblick

Für die Renyi-Dimensionsschätzer liegt auf einer endlichen Datenmenge keine Konvergenz der geschätzten Dimension gegen die echte Dimension für ϵ gegen 0 vor, und wir haben gesehen, dass durch den minimalen Abstand zweier Punkte in einem Datensatz immer eine untere Schranke für eine sinnvolle Messung gegeben ist. In diesem Zusammenhang ist die Verlässlichkeit der Schätzungen vor allem im grobskaligen Bereich $\epsilon \gg 0$ für weitere Beispiele zu untersuchen, um eventuell einen Zusammenhang zwischen der Attraktorgestalt und der Güte der Schätzung herzustellen. Wie wir beim Concentration of Measure-Experiment gesehen haben, ist dieser Bereich vor allem in hohen Dimensionen relevant.
Des Weiteren basiert das in dieser Arbeit verwendete Fehlerfunktional auf einem Maß für den Datenfehler, welches durch normalverteiltes Rauschen motiviert wurde. Hier wäre es interessant – analog zum Clustering-Algorithmus – andere Distanzbegriffe zu verwenden, welche zu anderen Arten von Datenstörungen korrespondieren. Analog ist es möglich Varianten der Hauptachsenzerlegung zu verwenden, die auf nicht-normalverteiltem Rauschen basieren, siehe [CDS01]. In diesem Zusammenhang wäre es sinnvoll zu untersuchen, ob mit der zum Rauschen passenden lokalen PCA ein besseres Resultat des Bregman-Clusterings mit den korrespondierenden Bregman-Divergenzen zu erreichen ist, als es in unseren Experimenten zu sehen war.
Weiterhin ist die Identifizierung eines praxisrelevanten Beispiels, für welches die H^1_{mix}-Regularisierung deutlich bessere Ergebnisse als die H^1-Regularisierung liefert, von Interesse.

Literaturverzeichnis

[AHK01] Aggarwal, C. C., A. Hinneburg und D. A. Keim: *On the Surprising Behavior of Distance Metrics in High Dimensional Space*. Seiten 420–434, 2001. Proceedings of the 8th International Conference on Database Theory.

[AK01] Antos, A. und I. Kontoyiannis: *Convergence Properties of Functional Estimates for Discrete Distributions*, 2001.

[ANN] `http://www.neural-forecasting-competition.com/NN3/index.htm`.

[Bel57] Bellman, R. E.: *Dynamic Programming*. Princeton University Press, 1957.

[BG04] Bungartz, H.-J. und M. Griebel: *Sparse grids*. Acta Numerica, 13:147–269, 2004.

[BK85] Broomhead, D. S. und G. P. King: *Extracting Qualitative Dynamics from Experimental Data*. Physica, D20:217–236, 1985.

[BMDG05] Banerjee, A., S. Merugu, I. S. Dhillon und J. Ghosh: *Clustering with Bregman Divergences*. Journal of Machine Learning Research, 6:1705–1749, 2005.

[BPZ08] Bungartz, H.-J., D. Pflüger und S. Zimmer: *Adaptive Sparse Grid Techniques for Data Mining*. 2008.

[Bra03] Braess, D.: *Finite Elemente*. Springer Verlag, 2003.

[Bra09] Braun, J.: *An Application of Kolmogorov's Superposition Theorem to Function Reconstruction in Higher Dimensions*. Doktorarbeit, Institut für Numerische Simulation, Universität Bonn, Dezember 2009.

[Cam03] Camastra, F.: *Data Dimensionality Estimation Methods: A Survey*. Pattern Recognition, 36(12):2945–2954, 2003.

[CDS01] Collins, M., S. Dasgupta und R. E. Schapire: *A Generalization of Principal Component Analysis to the Exponential Family*. In: *Advances in Neural Information Processing Systems*. MIT Press, 2001.

[Chi34] Chintchin, A.: *Korrelationstheorie der stationären stochastischen Prozesse*. Mathematische Annalen, 109:604–615, 1934.

[doC92] doCarmo, M. P.: *Riemannian Geometry*. Birkhäuser Boston, 1992.

[Eng91] Engel, U. M.: *Time Series Analysis*, 1991. A Part III Essay.

[ER92] Eckmann, J. P. und D. Ruelle: *Fundamental limitations for estimating dimensions and lyapounov exponents in dynamical systems*. Physica, D56:185–187, 1992.

[Feu10] Feuersänger, C.: *Sparse Grid Methods for Higher Dimensional Approximation*. Doktorarbeit, Institut für Numerische Simulation, Universität Bonn, 2010. Noch zu erscheinen.

[FG09] Feuersänger, C. und M. Griebel: *Principal Manifold Learning by Sparse Grids*. Computing, 85(4), 2009. Also available as INS Preprint no 0801.

[FS86] Fraser, A. M. und H. L. Swinney: *Independent coordinates for strange attractors from mutual information*. Physical Review A, 33(2):1134–1140, 1986.

[FSZ01] Fan, X., J. Shen und D. Zhao: *Sobolev Embedding Theorems for Spaces $W^{k,p(x)}(\Omega)$*. Journal of Mathematical Analysis and Applications, 262(2):749–760, 2001.

[Gal09] Galassi, M. et al: *GNU Scientific Library Reference Manual (3rd Ed.)*, 2009. ISBN 0954612078; `http://www.gnu.org/software/gsl/`.

[Gar04] Garcke, J.: *Maschinelles Lernen durch Funktionsrekonstruktion mit verallgemeinerten dünnen Gittern*. Doktorarbeit, Institut für Numerische Simulation, Universität Bonn, 2004.

[GG03] Gerstner, T. und M. Griebel: *Dimension–Adaptive Tensor–Product Quadrature*. Computing, 71(1):65–87, 2003.

[GGG09] Garcke, J., T. Gerstner und M. Griebel: *Intraday Foreign Exchange Rate Forecasting using Sparse Grids*. 2009. submitted.

[GH09] Garcke, J. und M. Hegland: *Fitting multidimensional data using gradient penalties and the sparse grid combination technique*. Computing, 84(1-2):1–25, 2009.

[GP83] Grassberger, P. und I. Procaccia: *Measuring the Strangeness of Strange Attractors*. Physica, D9:189–208, 1983.

[Gri98] Griebel, M.: *Adaptive Sparse Grid Multilevel Methods for elliptic PDEs based on finite differences.* Computing, 61(2):151–179, 1998.

[Gri06a] Griebel, M.: *Sparse grids and related approximation schemes for higher dimensional problems.* In: Pardo, L., A. Pinkus, E. Suli und M.J. Todd (Herausgeber): *Foundations of Computational Mathematics (FoCM05), Santander*, Seiten 106–161. Cambridge University Press, 2006.

[Gri06b] Griebel, M.: *Wissenschaftliches Rechnen I*, 2006. Skript zur Vorlesung am Institut für Numerische Simulation der Universität Bonn.

[Gru04] Gruene, L.: *Numerik Dynamischer Systeme*, 2004. Skript zur Vorlesung am Mathematischen Institut der Universität Bayreuth; `http://www.old.uni-bayreuth.de/departments/math/~lgruene/numdyn0304/`.

[GS73] Grams, W.F. und R. J. Serfling: *Convergence Rates for U-Statistics and Related Statistics.* The Annals of Statistics, 1(1):153–160, 1973.

[GSZ92] Griebel, M., M. Schneider und C. Zenger: *A combination technique for the solution of sparse grid problems.* In: Groen, P. de und R. Beauwens (Herausgeber): *Iterative Methods in Linear Algebra*, Seiten 263–281. IMACS, Elsevier, North Holland, 1992. also as SFB Bericht, 342/19/90 A, Institut für Informatik, TU München, 1990.

[GWST82] Greenside, H.S., A. Wolf, J. Swift und Pignataro T.: *Impracticality of a box-counting algorithm for calculating the dimensionality of strange attractors.* The American Physical Society, 25(6):3453–3456, 1982.

[HAK00] Hinneburg, A., C.C. Aggarwal und D.A. Keim: *What is the Nearest Neighbor in High Dimensional Spaces?* In: *Proceedings of the 26th VLDB Conference, Cairo*, Seiten 506–515, 2000.

[Heg03] Hegland, M.: *Adaptive Sparse Grids.* ANZIAM J., 44:C335–C353, 2003.

[Hen76] Henon, M.: *A Two-dimensional Mapping with a Strange Attractor.* Communications in Mathematical Physics, 50:69–77, 1976.

[HGC07] Hegland, M., J. Garcke und V. Challis: *The combination technique and some generalisations.* Linear Algebra and its Applications, 420(2–3):249–275, 2007.

[Hoe48] Hoeffding, W.: *A class of statistics with asymptotically normal distribution.* The Annals of Mathematical Statistics, 19(3):293–325, 1948.

[Hol08] Holtz, M.: *Sparse Grid Quadrature in High Dimensions with Applications in Finance and Insurance.* Doktorarbeit, Institut für Numerische Simulation, Universität Bonn, 2008.

[HP83] Hentschel, H. G. E. und I. Procaccia: *The infinite number of generalized dimensions of fractals and strange attractors.* Physica, D8:435–444, 1983.

[Huk06] Huke, J. P.: *Embedding Nonlinear Dynamical Systems: A Guide to Takens' Theorem*, 2006. Manchester Institute for Mathematical Sciences EPrint: 2006.26.

[Hun90] Hunt, F.: *Error Analysis and Convergence of Capacity Dimension Algorithms.* SIAM Journal of Applied Mathematics, 50(1):307–321, 1990.

[HZZGH82] Hong-Zhi, A., C. Zhao-Guo und E. J. Hannan: *Autocorrelation, Autoregression and Autoregressive Approximation.* The Annals of Statistics, 10(3):926–936, 1982.

[JL84] Johnson, W. und J. Lindenstrauss: *Extensions of Lipschitz maps into a Hilbert space.* Contemporary Mathematics, 26:189–206, 1984.

[KKKW95] Katayama, R., K. Kuwata, Y. Kajitani und M. Watanabe: *Embedding dimension estimation of chaotic time series using self-generating radial basis function network.* Fuzzy Sets and Systems, 71:311–327, 1995.

[Kru96] Krueger, A.: *Implementation of a fast box-counting algortihm.* Computer Physics Communications, 98:224–234, 1996.

[KS04] Kantz, H. und T. Schreiber: *Nonlinear Time Series Analysis.* Cambridge University Press, 2004. 2nd edition.

[Kus04] Kuska, J.-P.: *Mathematische Methoden der Bildverarbeitung*, 2004. Skript zur Vorlesung am Institut für Informatik der Universität Leipzig; `http://phong.informatik.uni-leipzig.de/~kuska/mathmeth.html`.

[LBG80] Linde, Y., A. Buzo und R. M. Gray: *An Algortihm for Vector Quantizer Design.* IEEE Transactions on Communications, COM-28(1):84–95, 1980.

[Lor63] Lorenz, E. N.: *Deterministic Nonperiodic Flow.* Journal of the Atmospheric Sciences, 20:130–141, 1963.

[Lou08] Loustau, S.: *Aggregation of SVM Classifiers Using Sobolev Spaces.* Journal of Machine Learning Research, 9:1559–1582, 2008.

[LT89] Liebovitch, L. S. und T. Toth: *A fast algorithm to determine fractal dimensions by box counting.* Physics Letters A, 141(8,9):386–390, 1989.

[LV07] Lee, J. A. und M. Verleysen: *Nonlinear Dimensionality Reduction.* Springer Science, 2007.

[Mai93] Mainieri, R.: *On the equality of Hausdorff and box counting dimensions.* AIP Chaos - An Interdisciplinary Journal of Nonlinear Science, 3(2):119–125, 1993.

[McG83] McGuinness, M. J.: *The fractal dimension of the Lorenz attractor.* Physics Letters A, 99:5–9, 1983.

[MSR+00] Müller, K.-R., A. J. Smola, G. Rätsch, B. Schölkopf, J. Kohlmorgen und V. Vapnik: *Using Support Vector Machines for Time Series Prediction*, 2000.

[Mur09] Murtaugh, F.: *The Remarkable Simplicity of Very High Dimensional Data: Application of Model-Based Clustering.* Journal of Classification, 26(3):249–277, 2009.

[MW94] Mikosch, T. und Q. Wang: *A Monte Carlo Method for Estimating the Correlation Exponent.* Journal of Statistical Physics, 78(3,4):799–813, 1994.

[Ott93] Ott, E.: *Measure and spectrum of D_q dimensions.* Chaos in Dynamical Systems, Seiten 78–81, 1993.

[PSVM92] Provenzale, A., L. A. Smith, R. Vio und G. Murante: *Distinguishing between low-dimensional dynamics and randomness in measured time series.* Physica, D58(31), 1992.

[Ren59] Renyi, A.: *On the dimension and entropy of probability distributions.* Acta Mathematica Hungarica, 10:193–215, 1959.

[Rue91] Ruelle, D.: *Chance and Chaos.* Princeton University Press, 1991.

[SS02] Schölkopf, B. und A. J. Smola: *Learning with Kernels – Support Vector Machines, Regularization, Optimization, and Beyond.* The MIT Press – Cambridge, Massachusetts, 2002.

[SSM96] Smola, A., B. Schölkopf und K.-R. Müller: *Nonlinear Component Analysis as a Kernel Eigenvalue Problem.* Technischer Bericht, MPI für biologische Kybernetik, Tübingen, 1996.

[SW98] Shannon, C. E. und W. Weaver: *The Mathematical Theory of Communication.* University of Illinois Press, 1998. reprinted.

[SYC91] Sauer, T., J. A. Yorke und M. Casdagli: *Embedology.* Journal of Statistical Physics, 65:576–616, 1991.

[Tak81] Takens, F.: *Detecting Strange Attractors In Turbulence.* Dynamical Systems and Turbulence, (898):366–381, 1981. Lecture Notes in Mathematics.

[The86] Theiler, J.: *Spurious dimension from correlation algorithms applied to limited time series data.* Physical Review A, 34:2427–2432, 1986.

[The87] Theiler, J.: *Efficient algorithm for estimating the correlation dimension from a set of discrete points.* Physical Review A, 36(9):4456–4462, 1987.

[Tik63] Tikhonov, A. N.: *Solution of Incorrectly Formulated Problems and the Regularization Method.* Soviet Math. Dokl., 4:1035–1038, 1963.

[VF05] Verleysen, M. und D. Francois: *The Curse of Dimensionality in Data Mining and Time Series Prediction.* Lecture Notes in Computer Science, 3512:758–770, 2005.

[WG92] Weigend, A. S. und N. A. Gershenfeld (Herausgeber): *Time Series Prediction: Forecasting the Future and Understanding the Past*, 1992. Proceedings of a NATO Advanced Research Workshop on Comparative Time Series Analysis, held in Santa Fe, New Mexico.