DIPLOMARBEIT

Numerische Berechnung der Transinformation von Zeitreihen

ausgeführt am Institut für

Nachrichtentechnik und Hochfrequenztechnik

der Technischen Universität Wien

von

Stefan Fröhlich

Arbesbachgasse 30/10

A-1190 Wien

29. Januar 1996

Diese Diplomarbeit wurde ausgeführt unter der Anleitung von
o.Univ.Prof. Dipl.-Ing. Dr. Hans Weinrichter und
Univ.Ass. Dipl.-Ing. Hans-Peter Bernhard

Kurzfassung

Diese Diplomarbeit befaßt sich mit der Untersuchung von zeit- und wertdiskreten Signalen, aus denen durch Zeitverzögerung Zustandsvektoren im Phasenraum erzeugt werden. Die auf diese Weise erzeugten Attraktoren weisen eine für das erzeugende System charakteristische Struktur auf. Sind die untersuchten Signale beispielsweise aufgezeichnete Vokale, so ist eine vom Sprecher weitgehend unabhängige Ähnlichkeit der Attraktoren zu bemerken. Im Rahmen dieser Arbeit wird versucht, mit numerischen Methoden die Ähnlichkeit solcher Strukturen zu erfassen, um auf diese Weise eine Klassifizierung der untersuchten Signale durchführen zu können. Im Bereich der Sprachverarbeitung wäre ein mögliches Anwendungsgebiet die Phonemseparation bzw. -klassifikation.

Um ein quantitatives Maß für die Ähnlichkeit zweier im Phasenraum eingebetteten Attraktoren zu erhalten, werden im weiteren Verlauf zwei verschiedene Größen verwendet, die Kullback Leibler Distanz und die Transinformation, die beide ihren Ursprung in der Informationstheorie haben. Zur Berechnung dieser Maße ist es notwendig, mehrdimensionale Wahrscheinlichkeitsdichteverteilungen zu schätzen. Für diese Aufgabe kommen verschiedene Verfahren in Frage, von denen drei im Rahmen dieser Arbeit näher untersucht werden. Es handelt sich dabei um das Histogramm mit adaptiver Klassengröße, um die Kerndichte Schätzung, sowie um eine Variante davon, den MMI–Algorithmus. Weiters wird das Histogramm mit fester Klassengröße zu Vergleichszwecken präsentiert, es eignet sich jedoch nicht für eine Schätzung der Wahrscheinlichkeitsdichteverteilung.

Nach einer kurzen Gegenüberstellung der theoretischen Eigenschaften folgt die Beschreibung der drei Algorithmen. Anschließend werden einige kurze Tests zur Verifikation der Ergebnisse durchgeführt. Dabei zeigt sich, daß das adaptive Histogramm Ergebnisse mit hoher Genauigkeit liefert. Die Kerndichteschätzung führt zu ähnlich exakten Ergebnissen, ist aber auf Grund der extrem langen Rechenzeiten für eine praktische Anwendung nicht geeignet. Durch Einsatz des aus der Literatur übernommenen MMI–Algorithmus kann die Laufzeit zwar gesenkt werden, jedoch sind die Ergebnisse falsch. Die Begründung für die auftretenden Fehler wird ebenfalls im Rahmen dieser Arbeit gegeben.

Der mit dem adaptiven Histogramm arbeitende Algorithmus kann für Berechnungen empfohlen werden, wenn die Rechengenauigkeit besonders wichtig ist. Für Anwendungen, bei denen die rasche Ausführung im Vordergrund steht, eignen sich die in [2] angegebenen Algorithmen besser.

Inhaltsverzeichnis

Kapitel 1

Einführung

In der digitalen Signalverarbeitung kam es in den letzten Jahren zu dramatischen technischen Verbesserungen, die unter anderem in neuen Produkten der Konsumelektronik ihren Niederschlag fanden. Die Fortschritte wurden sowohl im Bereich neuerer und vor allem schnellerer Hardware erzielt, als auch im theoretischen Bereich durch den Einsatz neu entwickelter Algorithmen. Auch die digitale Sprachverarbeitung, bei der es sich um ein Teilgebiet der Signalverarbeitung handelt, blieb von diesem Trend nicht ausgenommen.

In dieser Arbeit wird ein Ansatz für Aufgabenstellungen im Bereich der Phonemklassifizierung, basierend auf [1] untersucht, der bereits bei der Modellbildung neue Wege beschreitet. Bisher wurde für die Modellierung von menschlicher Sprache meist das Quelle-Filter-Modell verwendet. Dabei handelt es sich um eine lineare Nachbildung der menschlichen Sprachorgane. Ein Tongenerator, der die Stimmbänder simuliert, wird zur Erzeugung der stimmhaften Laute verwendet. Eine überlagerte Rauschquelle dient zur zur Nachbildung der Luftturbulenzen bei stimmlosen Lauten. Das resultierende Signal wird durch Filter geleitet, die die verschiedenen Resonanzräume im menschlichen Sprachapparat beschreiben. Bei genauerer Betrachtung zeigt sich jedoch, daß Sprachsignale zahlreiche Merkmale besitzen, die auf herkömmliche Weise nicht zufriedenstellend erklärt werden können. Wird zum Beispiel der selbe Laut von einem Sprecher mehrmals wiederholt, so beobachtet man bei jeder Realisierung einen völlig anderen Verlauf der Zeitfunktion.

Dies ist mit ein Grund, nichtlineare dynamische Systeme zur Beschreibung von Sprache einzusetzen. Das hier verwendete Modell geht dabei von der Annahme aus, daß das aufgezeichnete Sprachsignal durch einen Satz von gekoppelten, nichtlinearen Differentialgleichungen beschrieben werden kann.

Zur Beschreibung von Differentialgleichungssystemen wurde Mitte des 19. Jahrhunderts von Hamilton und Jacobi der Phasen- oder Zustandsraum entwickelt, mit dessen Hilfe sich auch dynamische nichtlineare Systeme beschreiben lassen. Dabei handelt es sich um einen mehrdimensionalen Raum, bei dem jede Koordinate einem Freiheitsgrad des untersuchten Systems entspricht. Aus einem n–dimensionalen Gleichungssystem erhält man zu jedem Zeitpunkt t einen n–dimensionalen Vektor als Lösung. Die Vektoren werden in einen Zustandsraum mit gleicher Ordnung wie das Gleichungssystem eingetragen. Auf diese Weise ergibt sich eine Kurve im Phasenraum, die durch t parame-

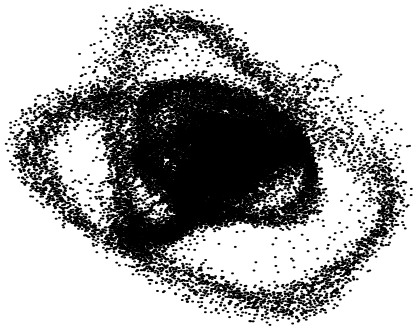
trisiert ist. Diese Kurve, die als Trajektorie bezeichnet wird, beschreibt das Verhalten des untersuchten Systems und ist die Lösung $x_0(t)$ für $t > 0$ mit $x_0 = x(0)$, also der Anfangsbedingung des Differentialgleichungssystems.

Untersucht man mit der oben beschriebenen Methode ein nichtlineares, dynamisches System im eingeschwungenen Zustand, so stellt man fest, daß sich die Trajektorie in einem abgegrenzten Bereich des Phasenraumes befindet. Das dadurch entstehende räumliche Gebilde wird als Attraktor des Systems bezeichnet. Der Attraktor gibt charakteristische Eigenschaften des zugrunde liegenden Systems wieder. Bei chaotischen Systemen kann man aus dem Attraktor auf Systemparameter wie die Ljapunov Exponenten und die Informationsproduktionsrate schließen. Bei Sprachsignalen ist eine Analyse weitaus schwieriger, da die Systemgleichungen nicht bekannt sind. Da nur ein eindimensionales Signal zur Verfügung steht, wird durch Zeitverzögerung ein mehrdimensionaler Vektor erzeugt, der dann in den Phasenraum eingetragen wird. Das Theorem von Takens sagt, daß $2n + 1$ Dimensionen für die Rekursion ausreichen, wenn das (in diesem Fall unbekannte) Gleichungssystem n–dimensional ist. Vergleicht man die Attraktoren mehrerer aufgezeichneter Vokale, so fällt auf, daß man intuitiv nach dem optischen Erscheinungsbild des Attraktors Zuordnungen treffen kann. Es stellt sich heraus, daß auch bei unterschiedlichen Sprechern die gleichen Vokale ähnliche Attraktoren im Phasenraum aufweisen.

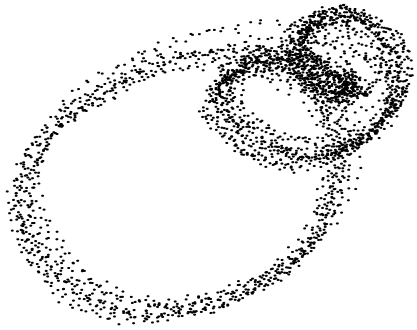
In den Abbildungen 1.1 und 1.2 sind solche Attraktoren dargestellt. Die ersten beiden Bilder zeigen die Vokale [a:] und [o:], gesprochen von Sprecher A. Dieselben Laute, diesmal von Sprecher B, sind in den weiteren beiden Bildern dargestellt. Man erkennt sofort die Ähnlichkeit der jeweils gleichen Vokale, während deutliche Unterschiede zwischen den Attraktoren für [a:] und denen für [o:] bestehen. Für weitergehende Betrachtungen wird auf die Diplomarbeit von Alfred Knapp verwiesen [9], die sich ausführlich dem Vergleich von Phasenportraits widmet.

Für eine Klassifizierung der Phasenportraits wäre es günstig, wenn die Ähnlichkeit, die vom menschlichen Betrachter leicht festgestellt werden kann, auch numerisch berechenbar wäre. Einen Ansatz dafür liefern die aus der Statistik stammenden Größen Kullback Leibler Distanz und Transinformation. Diese Maße werden verwendet, um Aussagen über die Ähnlichkeit zweier Wahrscheinlichkeitsdichteverteilungen treffen zu können. Da wir es bei unseren Problemen nicht unmittelbar mit Wahrscheinlichkeitsdichten zu tun haben, müssen die Attraktoren in Wahrscheinlichkeitsdichteverteilungen transformiert werden, um die Kullback Leibler Distanz bzw. die Transinformation anwenden zu können. Für unsere Berechnungen fassen wir den Phasenraum als mehrdimensionalen Ereignisraum auf, der für die numerische Berechnung in Klassen aufgeteilt wird. Die Trajektorie des untersuchten Systems beschreibt also die gesuchte Wahrscheinlichkeitsdichtefunktion im Ereignisraum. Da es sich um mehrdimensionale Daten handelt, sprechen wir von der Verbundwahrscheinlichkeitsdichte der einzelnen Koordinaten. In einer Klasse, die von der Trajektorie öfter durchlaufen wird, ist die Wahrscheinlichkeitsdichte höher als in einem Bereich, in dem nur wenige Durchläufe zu beobachten sind. Mit Hilfe der geschätzten Verbundwahrscheinlichkeitsdichte können die Transinformation und die Kullback Leibler Distanz berechnet werden.

Die Aufgabe dieser Diplomarbeit ist es, geeignete Programme zur Berechnung sowohl der Kullback Leibler Distanz als auch der Transinformation zur Verfügung zu

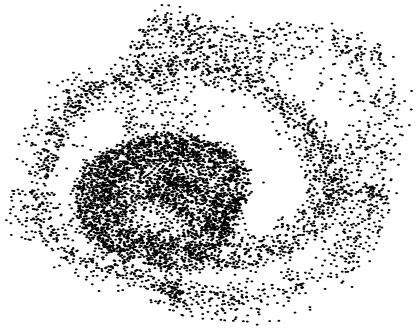


[a:]



[o:]

Abbildung 1.1: Sprecher A



[a:]

[o:]

Abbildung 1.2: Sprecher B

stellen. Dabei sind mehrere Algorithmen zu implementieren. Weiters ist ein Vergleich der erzielbaren Rechengenauigkeiten, eine Abschätzung der Ausführungsgeschwindigkeit und des jeweiligen Speicherbedarfs erforderlich.

Die entwickelten Algorithmen können in weiterer Folge in der Sprachverarbeitung zur Phonemseparation verwendet werden. Auch können mit Hilfe der Transinformation stationäre Abschnitte in Sprachsignalen erkannt werden. In der Systemtheorie sind die Algorithmen beispielsweise für die Bestimmung der Ljapunov-Exponenten eines unbekannten chaotischen Systems geeignet.

Im Kapitel 2 werden die mathematischen Grundlagen für die weitere Arbeit besprochen. Es wird beschrieben, wie aus den abgestasteten Zeitsignalen Phasenräume erzeugt werden. Weiters werden die Begriffe *Kullback-Leibler Distanz* und *Transinformation* definiert.

Im Kapitel 3 erfolgt eine Gegenüberstellung der untersuchten Berechnungsmethoden. Es wird sowohl das Prinzip der einzelnen Algorithmen als auch die konkrete Implementierung diskutiert. Weiters werden die Vor- und Nachteile der jeweiligen Verfahren einander gegenübergestellt.

Eine Zusammenfassung der Ergebnisse, die mit den entwickelten Programmen erhalten wurden, befindet sich im Kapitel 4. Es wird nicht nur die Genauigkeit des Ergebnisses bewertet, sondern auch die Ausführungsgeschwindigkeit und der Speicherbedarf. Zusätzlich zu den neu entwickelten Programmen werden auch bereits bestehende Algorithmen in den Vergleich einbezogen.

Die Bedienung der Programme wird in Kapitel 5 beschrieben. Es sind alle Optionen und Kommandozeilenparameter detailliert erklärt.

Kapitel 2

Grundlagen und Begriffsdefinitionen

2.1 Wahrscheinlichkeitsdichten mehrdimensionaler Zufallsgrößen

Bei den Untersuchungen, die im Rahmen dieser Diplomarbeit durchgeführt werden, kommen Verbundwahrscheinlichkeitsdichteverteilungen zum Einsatz. Diese werden aus Zeitreihen gebildet, wie im folgenden erläutert wird.

Ein zeit- und wertkontinuierliches Signal $s(t)$ wird n Mal mit der Zeitdauer T verzögert. Man erhält so die Signale $s(t), s(t-T), \ldots, s(t-(n-1)T)$. Diese werden als Vektor

$$\vec{s}_n^T(t) = [s(t), s(t-T), s(t-2T), \ldots, s(t-(n-1)T)] \tag{2.1}$$

angeschrieben. Aus Gründen der Übersichtlichkeit wird statt $\vec{s}_n^T(t)$ auch $\vec{s}_n^T$ geschrieben. Die Vektoren sind in einen n–dimensionalen Raum eingebettet. Eine graphische Übersicht über diesen Vorgang gibt Abbildung 2.1.

Die Amplitudendichtefunktion der Komponenten $s_i = s(t+iT)$ des Vektors $\vec{s}_n^T$ wird nun als Wahrscheinlichkeitsdichtefunktion $p_S(s_i)$ bezeichnet.

Im nächsten Schritt bildet man aus diesen skalaren Wahrscheinlichkeiten Verbundgrößen, und zwar

$$p(\vec{s}_n^T) = p(\vec{s}) = p\left(s\left(t\right), s\left(t-T\right), \ldots, s\left(t-\left(n-1\right)T\right)\right). \tag{2.2}$$

Über $p(\vec{s})$ läßt sich angeben, mit welcher Wahrscheinlichkeit der Vektor $\vec{s}_n^T$ zu einem beliebigen Zeitpunkt t in einem bestimmten Punkt des aufgespannten Raumes liegt.

Da bei einer numerischen Analyse zeit- bzw. wertkontinuierliche Signale nicht erfaßt werden können, verwenden wir in den entwickelten Programmen stets die amplitudendiskreten Abtastwerte $s(t)|_{t=kT}$. Auf diese Weise erhalten wir n–dimensionale Vektoren, mit deren Hilfe wir $p(\vec{s})$ schätzen müssen. Auf diese Problematik wird im Kapitel 3 im Detail eingegangen.

Selbstverständlich lassen sich aus den vektorwertigen Wahrscheinlichkeitsdichten wieder Verbundwahrscheinlichkeitsdichten erzeugen, man schreibt beispielsweise für

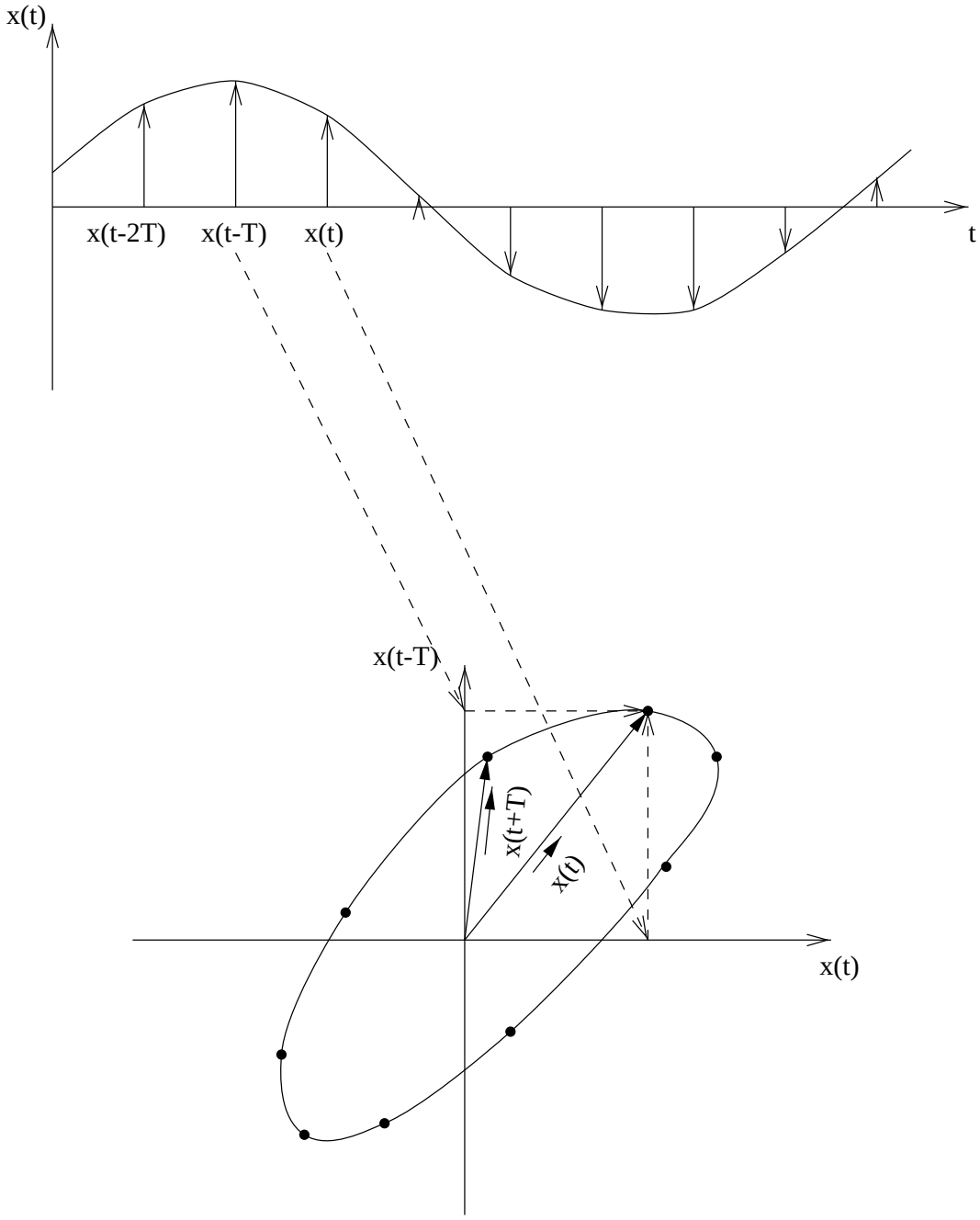


Abbildung 2.1: Konstruktion von Punkten im Phasenraum

das Verbundereignis, das aus den Signalen $x(t)$ und $y(t)$ gebildet wird

$$p_{XY}(\vec{x}_n^T, \vec{y}_n^T). \tag{2.3}$$

Wenn $\vec{x}$ die Dimension n und $\vec{y}$ die Dimension m hat, so ist die Verbundwahrscheinlichkeit in einen $(n+m)$–dimensionalen Raum eingebettet.

2.2 Entropie

Die Entropie ist eine Maßeinheit für den Informationsgehalt einer Zufallsvariablen. X sei eine diskrete Zufallsvariable aus dem Wahrscheinlichkeitsraum $\mathcal{X}$ mit einer Wahrscheinlichkeitsdichtefunktion $p_X(x)$. Die Entropie $H(X)$ dieser Zufallsvariablen ist der Erwartungswert des Informationsgehaltes[1]

$$\begin{aligned} H(X) &= E\{\log \frac{1}{p_X(x)}\} && (2.4)\\ &= -\sum_{x_n \in \mathcal{X}} p_X(x_n) \log p_X(x_n). && (2.5) \end{aligned}$$

Die Entropie ist unabhängig von einer konkreten Realisierung der Zufallsvariablen X, sondern ausschließlich eine Funktion der Verteilungsdichte $p_X(x)$. Außerdem ist die Entropie jeder Zufallsvariablen immer größer oder gleich Null,

$$H(X) \geq 0, \tag{2.6}$$

wobei $H(X) = 0$ genau dann auftritt, wenn eines der Elementarereignisse x_n von X das sichere Ereignis mit $p_X(x = x_n) = 1$ ist.

2.2.1 Entropie von Verbundereignissen

Die Definition für die Entropie aus dem vorigen Abschnitt kann ohne Probleme auf Verbundwahrscheinlichkeiten erweitert werden. Für die Verbundentropie $H(X,Y)$ gilt dann

$$H(X,Y) = -\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p_{X,Y}(x,y) \log p_{X,Y}(x,y). \tag{2.7}$$

Weiters kann man die bedingte Entropie als

$$H(Y|X) = -\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p_{X,Y}(x,y) \log p_{Y|X}(y|x) \tag{2.8}$$

definieren. Zwischen diesen beiden Größen gilt der Zusammenhang

$$H(X,Y) = H(X) + H(Y|X). \tag{2.9}$$

[1] Wenn für den Logarithmus die Basis 2 verwendet wird, dann hat das Ergebnis die Einheit *Bit*. In den weiteren Ausführungen wird, so es nicht explizit anders vermerkt ist, stets 2 als Basis für den Logarithmus angenommen.

2.2.2 Entropie von kontinuierlichen Zufallsvariablen

In (2.5) ist die Entropie als Summe über Elemente eines Wahrscheinlichkeitsraumes definiert. Damit ist die Entropie von diskreten Zufallsvariablen leicht zu berechnen, indem wir für jedes Elementarereignis ein Element definieren und anschließend die Summe bilden.

Bei kontinuierlichen Zufallsvariablen hingegen ist das nicht so einfach, weil keine geeigneten Partitionen definiert werden können (es gibt keine abzählbaren Elementarereignisse). Daher wird, um die Definition für die Entropie entsprechend zu erweitern, die *differentielle Entropie* verwendet.

X sei eine kontinuierliche Zufallsvariable mit der Wahrscheinlichkeitsdichtefunktion $f(x)$. Es gilt $f(x) > 0$ und $\int_{-\infty}^{\infty} f(x)\,dx = 1$. Die differentielle Entropie $h(X)$ dieser kontinuierlichen Zufallsvariable wird als

$$h(X) = -\int_X f(x) \log f(x)\,dx \tag{2.10}$$

definiert. Um den Zusammenhang zur Entropie diskreter Zufallsvariablen herzustellen, gehen wir folgendermaßen vor:

Die Wahrscheinlichkeitsdichtefunktion wird in gleich große Intervalle der Breite Δ aufgeteilt. Unter der Voraussetzung, daß die Dichteverteilung stetig ist, können wir für jedes Intervall einen Wert x_i bestimmen, für den die Gleichung

$$f(x_i)\Delta = \int_{i\Delta}^{(i+1)\Delta} f(x)\,dx \tag{2.11}$$

erfüllt ist.
Definieren wir den Wertebereich eines Quantisierungsintervalles der Zufallsvariable

$$X^\Delta = x_i, \quad \text{wenn} \quad i\Delta \le X < (i+1)\Delta, \tag{2.12}$$

so läßt sich die Wahrscheinlichkeit für das Ereignis $X^\Delta = x_i$ als

$$p_i = \int_{i\Delta}^{(i+1)\Delta} f(x)\,dx = f(x_i)\Delta \tag{2.13}$$

anschreiben.
Die Entropie der quantisierten Zufallsvariable folgt aus (2.5) als

$$\begin{aligned} H(X^\Delta) &= -\sum_{-\infty}^{\infty} p_i \log p_i && (2.14)\\ &= -\sum_{-\infty}^{\infty} f(x_i)\Delta \log\left(f\left(x_i\right)\Delta\right) && (2.15)\\ &= -\sum_{-\infty}^{\infty} \Delta f(x_i) \log f(x_i) - \log\Delta \sum_{-\infty}^{\infty} f(x_i)\Delta && (2.16)\\ &= -\sum_{-\infty}^{\infty} \Delta f(x_i) \log f(x_i) - \log\Delta. && (2.17) \end{aligned}$$

Dabei wird ausgenützt, daß

$$\sum_{-\infty}^{\infty} f(x_i)\Delta = 1. \tag{2.18}$$

Für $\Delta \to 0$ gilt

$$\sum_{-\infty}^{\infty} \Delta f(x_i) \log f(x_i) \quad \to \quad \int_X f(x) \log f(x)\, dx \tag{2.19}$$

falls $f(x) \log f(x)$ Riemann-integrierbar ist. Als Zusammenhang zwischen der Entropie von kontinuierlichen und diskreten Zufallsvariablen können wir daher die Beziehung

$$h(X) = H(X^{\Delta}) + \log \Delta \tag{2.20}$$

angeben.

2.3 Kullback Leibler Distanz und Transinformation

Nachdem wir die Entropie als den Informationsgehalt einer Zufallsvariablen eingeführt haben, werden als nächster Schritt Größen definiert, mit denen Aussagen über den Zusammenhang zwischen zwei Wahrscheinlichkeitsverteilungen getroffen werden können. In dieser Arbeit werden zwei solche Maße untersucht, die *Kullback Leibler Distanz* und die *Transinformation*.

2.3.1 Kullback Leibler Distanz

Die Kullback Leibler Distanz $D(p \parallel q)$ wird als Abstandsmaß zum Vergleich zweier Zufallsvariablen verwendet. Wenn etwa die Verteilung $q_X(x)$ gegeben ist, so gibt $D(p \parallel q)$ an, wie gut eine unbekannte Verteilung $p_X(x)$ mit $q_X(x)$ geschätzt werden kann.

Die Definition der Kullback Leibler Distanz lautet

$$D(p \parallel q) = \sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q(x)} \tag{2.21}$$

$$= E_p \log \frac{p(X)}{q(X)}, \tag{2.22}$$

wobei E_p der Erwartungswert der Verteilung $p(x)$ ist.

In der obigen Definition werden die Verteilungen als im ganzen Raum positiv definiert

$$p_X(x) > 0, \quad q_X(x) > 0. \tag{2.23}$$

Diese Bedingungen können auf

$$p_X(x) \geq 0, \quad q_X(x) \geq 0. \tag{2.24}$$

erweitert werden, wenn man die Definitionen

$$0 \log \frac{0}{q_X(x)} = 0 \tag{2.25}$$

und

$$p_X(x) \log \frac{p_X(x)}{0} = \infty \tag{2.26}$$

einführt.

Wie die Entropie ist auch die Kullback Leibler Distanz nie negativ, wobei

$$D(p \parallel q) = 0 \tag{2.27}$$

dann und nur dann gilt wenn $p_X(x) = q_X(x)$. Dennoch handelt es sich um kein echtes Abstandsmaß, weil einerseits keine Symmetrie bezüglich $p_X(x)$ und $q_X(x)$ vorliegt und andererseits auch die Dreiecksungleichung nicht erfüllt ist.

Gilt für einen beliebigen Punkt x aus X

$$p_X(x) \neq 0 \quad \text{und} \quad q_X(x) = 0, \tag{2.28}$$

so ist die Kullback Leibler Distanz unendlich groß, unabhängig von der Wahrscheinlichkeitsdichteverteilung im Rest des Wahrscheinlichkeitsraumes. Dadurch wird eine Interpretation des Ergebnisses unmöglich. Deshalb muß bei der Berechnung der Kullback Leibler Distanz

$$q_X(x) > 0 \quad \forall x \tag{2.29}$$

gefordert werden. Auf die Konsequenzen dieser Bedingung wird im Kapitel 3.2.3 bei der Diskussion des Algorithmus näher eingegangen.

2.3.2 Transinformation

Die Transinformation ist ein Spezialfall der Kullback Leibler Distanz. Gegeben sind zwei Zufallsvariablen X und Y sowie deren Verbundwahrscheinlichkeitsdichte $p_{X,Y}(x,y)$ und die beiden Randverteilungen $p_X(x)$ und $p_Y(y)$. Die Transinformation $I(X;Y)$ läßt sich dann aus

$$\begin{aligned} I(X;Y) &= \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p_{X,Y}(x,y) \log \frac{p_{X,Y}(x,y)}{p_X(x) p_Y(y)} && (2.30) \\ &= D(p_{X,Y}(x,y) \parallel p_X(x) p_Y(y)) && (2.31) \\ &= E_{p(X,Y)} \log \frac{p(X,Y)}{p(X) p(Y)} && (2.32) \end{aligned}$$

berechnen. Die Transinformation ist daher aus der Kullback Leibler Distanz ableitbar. Sie gibt Auskunft darüber, wieviel Information über $p_X(x)$ in $p_Y(y)$ enthalten ist. Im Gegensatz zur Kullback Leibler Distanz gilt

$$I(X;Y) = I(Y;X), \tag{2.33}$$

die Transinformation ist kommutativ bezüglich X und Y.

Zwischen der Transinformation $I(X;Y)$ und den Entropien $H(X)$ und $H(Y)$ gelten folgende Zusammenhänge:

$$I(X;Y) = H(X) - H(X|Y) \tag{2.34}$$

und daher auf Grund der Symmetriebedingungen auch

$$I(X;Y) = H(Y) - H(Y|X). \tag{2.35}$$

Weiters gilt

$$I(X;Y) = H(X) + H(Y) - H(X,Y) \tag{2.36}$$

und

$$I(X;X) = H(X). \tag{2.37}$$

Die letzte Gleichung besagt, daß die Information einer Zufallsvariablen über sich selbst identisch mit der Entropie dieser Zufallsvariablen ist.

Die Gleichung (2.34) ist wichtig, wenn die Transinformation von kontinuierlichen Zufallsvariablen berechnet werden soll. Setzt man in

$$I(X;Y) = h(X) - h(X|Y) \tag{2.38}$$

für die beiden Entropien (2.20) ein, so erhält man

$$I(X;Y) = H(X) + \log \Delta_1 - H(X|Y) - \log \Delta_2. \tag{2.39}$$

Unter der Bedingung, daß $\Delta_1 = \Delta_2$ führt (2.39) wieder auf (2.34) zurück. Es läßt sich die Transinformation kontinuierlicher Zufallsvariablen über die quantisierten Wahrscheinlichkeitsdichteverteilungen berechnen. Die Voraussetzung dafür ist, daß die Quantisierungsintervalle geeignet gewählt werden.

2.3.3 Erweiterung auf vektorwertige Zufallsvariablen

Nachdem bereits Verbundwahrscheinlichkeitsdichten für vektorwertige Zufallsvariablen definiert wurden, ist die Erweiterung der Begriffe *Transinformation* und *Kullback Leibler Distanz* für solche Größen nicht mehr schwierig. Die Transinformation $I(X;Y)$ läßt sich als

$$I(X;Y) = \sum_{\vec{x}\in\mathcal{X}} \sum_{\vec{y}\in\mathcal{Y}} p(\vec{x}_n, \vec{y}_n) \log \frac{p(\vec{x}_n, \vec{y}_n)}{p(\vec{x}_n)p(\vec{y}_n)} \tag{2.40}$$

anschreiben, für die Kullback Leibler Distanz ergibt sich

$$D(\vec{p} \parallel \vec{q}) = \sum_{\vec{x}\in\mathcal{X}} p(\vec{x}_n) \log \frac{p(\vec{x}_n)}{q(\vec{x}_n)}. \tag{2.41}$$

Kapitel 3

Berechnungsverfahren für Transinformation und Kullback Leibler Distanz

Um die Transinformation oder die Kullback Leibler Distanz einer Zeitreihe zu berechnen, muß die Wahrscheinlichkeitsdichtefunktion bestimmt werden, die durch Abbildung der Meßpunkte im Phasenraum entsteht. Die wesentliche Aufgabe der Algorithmen besteht daher darin, aus den einzelnen, zeitlich aufeinanderfolgenden Abtastwerten eine Schätzung für die Wahrscheinlichkeitsdichteverteilung der Meßpunkte im Phasenraum durchzuführen. Für dieses Problem kommen mehrere Verfahren in Frage, nämlich das Histogramm, sowohl mit fester als auch mit variabler Klassengröße, sowie die Kerndichte Schätzung.

3.1 Histogramm–Methode

Die bekannteste und älteste Methode zum Schätzen von Wahrscheinlichkeitsdichten ist die sogenannte Histogrammethode, die Ende des 19. Jahrhunderts von Pearson entwickelt wurde [12].

3.1.1 Prinzip

Um ein Histogramm zu erzeugen, wird nach der folgenden Methode vorgegangen:

Zunächst werden N gleich große disjunkte Intervalle gebildet, die in ihrer Gesamtheit den Wertebereich der Eingangsdaten vollständig abdecken. Die einzelnen Datenpunkte werden auf Grund ihres Wertes einem dieser Intervalle zugeordnet. Anschließend wird gezählt, wieviele Punkte in jedem Intervall enthalten sind. Die Anzahl der Punkte (sie wird auch *absolute Häufigkeit* genannt) wird nun durch die Gesamtzahl der Datenpunkte dividiert. Man erhält so die *relative Häufigkeit* als diskrete Wahrscheinlichkeitsverteilung. Diese kann als geschätzte Funktion der kontinuierlichen Wahrscheinlichkeitsdichtefunktion verwendet werden.

3.1.2 Wahl der Intervallgröße

Für das Schätzen der Wahrscheinlichkeitsdichtefunktion spielt die Wahl der Intervallgröße eine zentrale Rolle. Betrachten wir als Beispiel eine Sinusschwingung $x(t)$, der gleichverteiltes Rauschen $n(t)$ mit einem Signal-Rausch-Abstand von 42 dB additiv überlagert ist, wobei der Signal-Rausch-Abstand als

$$\mathrm{SNR} = \frac{\int_{\infty}^{\infty} x^2 p_X(x)\,dx}{\int_{\infty}^{\infty} n^2 p_N(n)\,dn} \tag{3.1}$$

definiert ist.

Die Amplitudendichteverteilung der Sinusfunktion ist

$$p_{sin}(x) = \frac{1}{\pi}\frac{1}{\sqrt{1-x^2}}, \tag{3.2}$$

das Rauschen wird durch die Rechteckfunktion

$$p_{noise}(x) = \begin{cases} \frac{1}{b-a} & : \quad b < x < a \\ 0 & : \quad \text{sonst} \end{cases} \tag{3.3}$$

beschrieben. Die Grenzen a und b sind so gewählt, daß der gewünschte Signal-Rausch Abstand von 42 dB erreicht wird.

Aus dem unverrauschten Signal $s_{orig}(t)$ wird durch Überlagerung mit dem Rauschsignal $n(t)$ das Summensignal $s(t) = s_{orig}(t) + n(t)$ gebildet Um die Verteilungsdichte dieses Summensignals zu erhalten, muß die Faltung der beiden Verteilungsdichtefunktionen gebildet werden,

$$p(x) = p_{sin}(x) * p_{noise}(x) \tag{3.4}$$

$$= \int_{-\infty}^{\infty} p_{sin}(\tau) \cdot p_{noise}(x-\tau)\,d\tau. \tag{3.5}$$

Man erhält nach kurzer Zwischenrechnung

$$p(x) = \begin{cases} 0 & : \quad x < -1 - \frac{b-a}{2} \\ \frac{1}{\pi(b-a)}\left(\arcsin\left(x + \frac{b-a}{2}\right) + \frac{\pi}{2}\right) & : \quad -1 - \frac{b-a}{2} < x < -1 + \frac{b-a}{2} \\ \frac{1}{\pi(b-a)}\left(\arcsin\left(x + \frac{b-a}{2}\right) - \arcsin\left(x - \frac{b-a}{2}\right)\right) & : \quad -1 + \frac{b-a}{2} < x < 1 - \frac{b-a}{2} \\ \frac{1}{\pi(b-a)}\left(\frac{\pi}{2} - \arcsin\left(x - \frac{b-a}{2}\right)\right) & : \quad 1 - \frac{b-a}{2} < x < 1 + \frac{b-a}{2} \\ 0 & : \quad 1 + \frac{b-a}{2} < x. \end{cases} \tag{3.6}$$

Diese Funktion wird in Abbildung 3.1 dargestellt.

Betrachten wir zum Schätzen der Wahrscheinlichkeitsdichtefunktion eine Stichprobe von 800 Abtastwerten und erzeugen ein Histogramm mit 10 Klassen. Es ergibt sich eine Darstellung wie in Bild 3.2.

Durch die geringe Anzahl an Intervallen ist es unmöglich, den exakten Verlauf der Wahrscheinlichkeitsdichteverteilung zu erfassen. Lokale Schwankungen, die kleiner sind als die Intervallbreite, werden auf Grund der breiten Intervalle nicht aufgelöst.

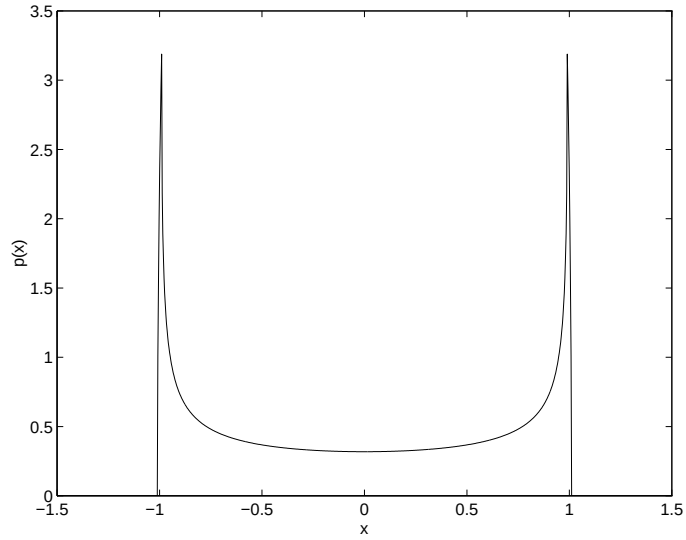


Abbildung 3.1: Mathematisch berechnete Amplitudendichte einer Sinusfunktion mit additivem Rauschen

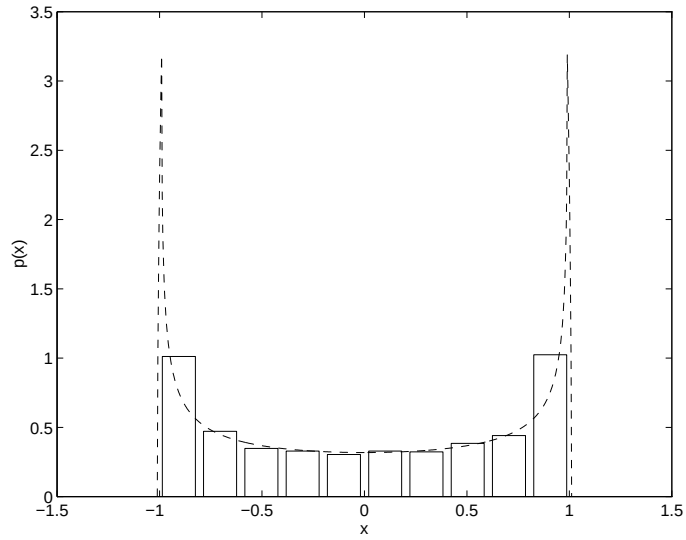


Abbildung 3.2: Histogramm mit 10 Intervallen

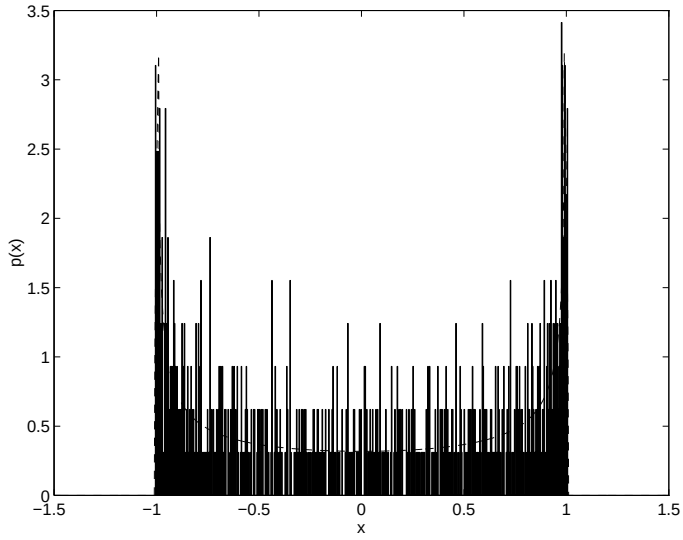


Abbildung 3.3: Histogramm mit 500 Intervallen

Um eine Abschätzung für die Güte der Schätzung zu erhalten, verwenden wir den absoluten quadratischen Fehler ϵ^2, der als

$$\epsilon^2 = \int_{-\infty}^{\infty} \left(\hat{p}\left(x\right) - p\left(x\right)\right)^2 \, dx \tag{3.7}$$

definiert ist. Mit $\hat{p}(x)$ wird der Schätzwert der Wahrscheinlichkeitsdichtefunktion an der Stelle x bezeichnet, $p(x)$ steht für den entsprechenden analytischen Wert.

Für das Histogramm mit 10 Klassen beträgt der quadratische Fehler nach (3.7) $\epsilon^2 = 137 \cdot 10^{-3}$.

Erhöht man bei der Erstellung des Histogrammes die Anzahl der Intervalle auf 500, so erhält man ein Resultat, wie es in Bild 3.3 dargestellt ist. In der Abbildung können wir nun zwar lokale Unregelmäßigkeiten der Wahrscheinlichkeitsdichte erkennen, allerdings handelt es sich dabei um einen Verlauf, der durch die willkürliche Lage der Intervallgrenzen und die zufällige Phasenlage der Sinusfunktion erzeugt wurde. Weiters stellt man fest, daß viele Intervalle nur noch einen oder zwei Datenpunkte enthalten, somit ist auch die vertikale Auflösung entsprechend gering: es kommt zu einer Quantisierung mit unzureichender Menge an Datenpunkten. Der quadratische Fehler nach (3.7) beträgt für dieses Histogramm $\epsilon^2 = 609 \cdot 10^{-3}$.

Bei einer Anzahl von 35 äquidistanten Teilbereichen ergibt sich für unser Beispiel ein Histogramm, das die zu schätzende Wahrscheinlichkeitsdichte sehr gut wiedergibt (Bild 3.4). Der quadratische Fehler ist in diesem Fall $\epsilon^2 = 52 \cdot 10^{-3}$, also deutlich geringer als in den beiden vorangegangenen Beispielen.

Die Problematik bei der Wahl der Intervallgröße ist, daß man die Struktur der Wahrscheinlichkeitsdichte kennen müßte, um die Anzahl an Intervallen festlegen zu können. Es sind also gewisse Kenntnisse über die gesuchte Größe notwendig, um Histogramme verwenden zu können.

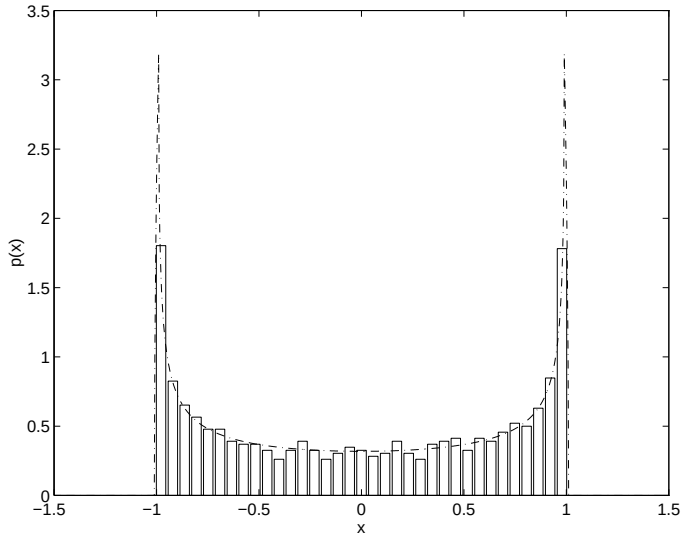


Abbildung 3.4: Histogramm mit 35 Intervallen

Es gibt einige allgemeine Richtlinien zur Erstellung von Histogrammen [12], es soll hier nur die Regel von Sturges erwähnt werden, die bei N Datenpunkten

$$k = 1 + \log_2 N \tag{3.8}$$

Intervalle empfiehlt. Die Grenzen solcher generellen Empfehlungen zeigen sich bereits am bisher betrachteten Beispiel der verrauschten Sinusschwingung. Nach der Regel von Sturges würden etwa 10 Intervalle für eine Beschreibung ausreichen. Durch die Fehlerabschätzung wurde aber gezeigt, daß die dreifache Anzahl an Intervallen ein wesentlich exakteres Ergebnis bringt. Die Begründung für diese Abweichung liegt darin, daß die zu schätzende Wahrscheinlichkeitsdichtefunktion keine Ähnlichkeit mit der Binomialverteilung aufweist, aus der die angeführte Regel abgeleitet wurde.

Wie gezeigt wurde, muß die Intervallgröße in Abhängigkeit von der Anzahl der Datenpunkte gewählt werden. Befinden sich die Daten auf wenigen lokal begrenzten Bereichen des untersuchten Raumes, während andere Teile nur sehr wenige Werte enthalten, so ist es nicht mehr möglich, geeignete äquidistante Intervalle zu wählen. Der Grund dafür ist, daß sie zu klein für die schwach besetzten Teile des Raumes sind, falls die Teilungen auf die dicht gefüllten Bereiche abgestimmt sind. Paßt man die Intervallgrößen an die Bereiche mit geringer Wahrscheinlichkeitsdichte an, so sind sie für die dicht besetzten Bereiche zu grob gewählt.

Dieser unregelmäßige Verlauf der Wahrscheinlichkeitsdichte tritt mit steigender Dimension oder abnehmender Anzahl an Datenpunkten stärker in Erscheinung.

Bei Histogrammen im n–mensionalen Raum muß jede Koordinate entsprechend der Intervallgröße geteilt werden. Es ergeben sich daher bei K Unterteilungen in n Dimensionen K^n Intervalle. Die Anzahl der Intervalle (und bei einer numerischen Auswertung somit auch die Rechenzeit) steigt dadurch exponentiell mit der Anzahl der Dimensionen.

Auf Grund dieser Nachteile kann das Histogramm nicht für die gegebenen Problemstellungen verwendet werden. Es wurde daher auf eine Implementierung verzichtet.

Um die angeführten Schwierigkeiten zu vermeiden, bieten sich zwei weitere Methoden zur Schätzung der Wahrscheinlichkeitsdichteverteilung an, das *adaptive Histogramm* und die *Kerndichte Schätzung.*

3.2 Adaptive Histogramm–Methode

3.2.1 Schätzung der Wahrscheinlichkeitsdichteverteilung

3.2.1.1 Prinzip

Beim adaptiven Histogramm handelt es sich um eine Verbesserung der Methode, wie sie im Abschnitt 3.1 besprochen wurde. Als einer der wesentlichen Nachteile des Histogramms wurde dort die feste, im vorhinein bestimmte Intervallgröße erkannt. Ein adaptives Histogramm dagegen ist ein Histogramm, dessen einzelne Intervalle nicht die gleiche Größe besitzen.

Man kann eine Verbesserung der Schätzgenauigkeit erzielen, indem man die Intervallbreiten lokal an die Wahrscheinlichkeitsdichteverteilung anpaßt. Um solch ein Histogramm zu erzeugen, gibt es zwei unterschiedliche Lösungsansätze:

- Man erzeugt zunächst ein Histogramm, wie es in Kapitel 3.1 beschrieben wurde. Dabei wählt man die Intervalle so eng, daß in den Bereichen mit hoher Wahrscheinlichkeitsdichte eine gute Anpassung an die Datenstruktur erreicht wird.

  In den Bereichen, die vorwiegend leere Intervalle enthalten, legt man nun zwei oder mehrere benachbarte Intervalle zu einem größeren Intervall zusammen. Für die Anzahl der Intervalle, die zusammengelegt werden gelten dabei dieselben Kriterien, die für die Intervallgöße in Kapitel 3.1.2 diskutiert wurden.

  Das resultierende adaptive Histogramm besteht aus Intervallen, die alle die gleiche Wahrscheinlichkeit besitzen, aber unterschiedliche Breite aufweisen. Die resultierende Wahrscheinlichkeitsdichte kann durch Division der Wahrscheinlichkeit durch die Intervallbreite errechnet werden.

- Um nicht zuerst ein Histogramm mit fester Intervallgröße erzeugen zu müssen, das zu einem adaptiven Histogramm umgewandelt wird, gehen wir in dieser Arbeit einen anderen Weg.

  Zunächst werden besonders breite Intervalle gewählt, meist für jede Koordinate eine oder gar keine Unterteilung. Wir betrachten ein einzelnes Intervall und untersuchen, ob darin Gleichverteilung vorliegt, oder ob die Wahrscheinlichkeitsdichte noch Strukturen aufweist. Für diese Prüfung muß ein geeignetes Kriterium zur Verfügung gestellt werden; auf diese Frage wird in Kapitel 3.2.1.3 eingegangen.

  Liegt Gleichverteilung vor, so ist eine weitere Unterteilung nicht nötig (der Beweis dafür wird im folgenden Kapitel geführt). Existieren noch Feinstrukturen, so teilt man das Intervall in (im Prinzip beliebige) kleinere Intervalle auf und wiederholt den Test auf Gleichverteilung für jedes Teilgebiet.

3.2.1.2 Bestimmung der Intervallegröße

Nachdem das Histogramm mit fester Intervallgröße ein Spezialfall des adaptiven Histogramms ist, findet man zu jedem Histogramm ein adaptives Histogramm, das zumindest gleich gut für eine bestimmte Aufgabe geeignet ist. Allerdings gibt es kein Optimalitätskriterium, mit dem adaptive Histogramme erzeugt werden können.

Im eindimensionalen Fall erweist es sich nach [12], Seite 71, als günstig, die Intervallgröße indirekt proportional zur Wahrscheinlichkeitsdichte festzulegen. Das führt dazu, daß in jedem Intervall annähernd gleich viele Datenpunkte enthalten sind. Dadurch wird eine Anpassung an lokale Datenstrukturen gewährleistet.

Da es im n–dimensionalen Raum nicht mehr möglich ist, disjunkte Hyperkuben so zu wählen, daß alle gleich viele Datenpunkte enthalten, ist dieses Kriterium für unsere Anwendung nicht geeignet. Es wird daher das im vorigen Abschnitt beschriebene Verfahren eingesetzt, bei dem versucht wird, die Intervalle so zu wählen, daß innerhalb stets eine Gleichverteilung vorliegt. Die Intervallgröße steht dabei in keinem Zusammenhang mit der Anzahl der enthaltenen Datenpunkte.

Bisher haben wir stillschweigend angenommen, daß Intervalle, in denen die zu schätzende Wahrscheinlichkeitsdichtefunktion konstant ist, für unsere Berechnungen geeignet sind. Im folgenden wird diese Annahme bewiesen.

Postulieren wir zu diesem Zweck: ein Element $\overrightarrow{xy}_i$ des Verbundwahrscheinlichkeitsraumes $\mathcal{XY}$ ist dann optimal gewählt, wenn die Wahrscheinlichkeitsdichte im gesamten Intervall konstant $p(\vec{x},\vec{y})$ ist. Berechnen wir nun den Beitrag $I_i(X;Y)$ des Intervalles $\overrightarrow{xy}_i$ aus $\mathcal{XY}$ zur Transinformation $I(X;Y)$. Wir teilen $\vec{x}_i$ in L und $\vec{y}_i$ in M beliebige Unterbereiche auf und erhalten so $K = LM$ Unterbereiche im Raum der Verbundwahrscheinlichkeitsdichte. Dabei können jedem Bereich $\overrightarrow{xy}_{lm}$ genau definierte Intervalle $\vec{x}_l$ und $\vec{x}_m$ für die Berechnung der Randverteilungen zugeordnet werden. Wir berechnen nun für jeden Unterbereich $\overrightarrow{xy}_{lm}$ den Beitrag $I_{i_{lm}}(X;Y)$ zur Transinformation. Dabei werden mit $\vec{x}_{i_l}$ und $\vec{x}_{i_m}$ immer die Werte der Randverteilungen bezeichnet, die dem Laufindex der Verbundwahrscheinlichkeit zuzuordnen sind.

Da die Wahrscheinlichkeitsdichten in $\vec{x}_i$ und $\vec{y}_i$ sowie die Verbundwahrscheinlichkeitsdichte $p(\vec{x}_i,\vec{y}_i)$ konstant sind, gilt:

$$a_l = \frac{p(\vec{x}_{i_l})}{p(\vec{x}_i)} \quad 1 \le l \le L, \tag{3.9}$$

$$b_m = \frac{p(\vec{y}_{i_m})}{p(\vec{y}_i)} \quad 1 \le m \le M \tag{3.10}$$

und

$$c_{lm} = \frac{p(\vec{x}_{i_l},\vec{y}_{i_m})}{p(\vec{x}_i,\vec{y}_i)} \quad 1 \le lm \le K, \tag{3.11}$$

wobei c_{lm} den Anteil des Unterbereiches lm am Hypervolumen des Bereiches i angibt, für a_l und b_m gilt dies sinngemäß. Außerdem gilt die Beziehung

$$c_{lm} = a_l b_m. \tag{3.12}$$

Die Transinformation für den Unterbereich lm kann daher mit

$$I_{i_{lm}}(X;Y) = p(\vec{x}_{i_l}, \vec{y}_{i_m}) \log \frac{p(\vec{x}_{i_l}, \vec{y}_{i_m})}{p(\vec{x}_{i_l})p(\vec{y}_{i_m})} \tag{3.13}$$

$$= c_{lm} p(\vec{x}_i, \vec{y}_i) \log \frac{c_{lm} p(\vec{x}_i, \vec{y}_i)}{a_l p(\vec{x}_i) b_m p(\vec{y}_i)} \tag{3.14}$$

$$= c_{lm} p(\vec{x}_i, \vec{y}_i) \log \frac{p(\vec{x}_i, \vec{y}_i)}{p(\vec{x}_i)p(\vec{y}_i)} \tag{3.15}$$

geschrieben werden. Die Summe über alle Unterbereiche ergibt den Ausdruck

$$I_i(X;Y) = p(\vec{x}_i, \vec{y}_i) \log \frac{p(\vec{x}_i, \vec{y}_i)}{p(\vec{x}_i)p(\vec{y}_i)} \sum_{lm=1}^{K} c_{lm}. \tag{3.16}$$

Da die Summe der Unterräume den ursprünglichen Bereich ergibt, gilt außerdem

$$\sum_{lm=1}^{K} p(\vec{x}_{i_l}, \vec{y}_{i_m}) = p(\vec{x}_i, \vec{y}_i) \tag{3.17}$$

und daher auch

$$\sum_{lm=1}^{K} c_{lm} = 1. \tag{3.18}$$

Damit reduziert sich $I_i(X;Y)$ auf den i–ten Summanden in (2.30). Der Beitrag eines Bereiches zur Transinformation ist demnach unabhängig von einer weiteren Unterteilung, wenn Gleichverteilung vorliegt.

Daraus folgt: *Ein Ausschnitt aus dem als Wahrscheinlichkeitsraum aufgefaßten Phasenraum ist dann als Unterbereich für die Berechnung der Transinformation geeignet, wenn in seinem Inneren alle Verteilungsfunktionen konstant sind.*

Diese Richtlinie schreibt vor, wie weit der Phasenraum unterteilt werden muß, um eine genaue Berechnung zu ermöglichen. Es wird allerdings noch keine konkrete Regel angegeben, auf welche Art diese Unterteilungen durchzuführen sind. Im Prinzip kann ein Intervall in eine beliebige Anzahl an Subintervallen unterteilt werden. Die Subintervalle dürfen dabei jede beliebige Form besitzen. Die Wahl eines bestimmten Verfahrens wird erst bei der Implementation des Algorithmus getroffen.

3.2.1.3 Das χ^2–Kriterium

Um feststellen zu können, ob im betrachteten Abschnitt des Phasenraumes Gleichverteilung vorliegt, wird ein geeignetes Vergleichskriterium benötigt. Da, bedingt durch die endliche Anzahl an Datenpunkten, statistische Schwankungen in der Wahrscheinlichkeitsdichteverteilung auftreten, muß eine Toleranzschwelle verwendet werden, die bestimmt, wann eine gegebene Verteilung als gleichverteilt betrachtet wird und ab wann Feinstrukturen in der zu schätzenden Funktion vermutet werden.

Da die Verwendbarkeit des Algorithmus in hohem Ausmaß von der geeigneten Wahl des Abbruchkriteriums abhängt, kommt diesem eine entsprechend hohe Bedeutung zu. Einerseits darf möglichst keine existierende Unterstruktur übersehen werden, da sonst

Information verloren ginge. Andererseits darf nicht der Fehler gemacht werden, Strukturen, die auf den Abtastzeitpunkt zurückzuführen sind, als wesentlich zu interpretieren, da sonst der Informationsgehalt überschätzt würde. Besonders der zweite Fall tritt bei höherdimensionalen Berechnungen stärker auf, da dann selbst große Punktemengen (mehrere 10^5 Punkte) nicht mehr ausreichen, den Raum genügend dicht auszufüllen. Die weit voneinander entfernten Datenpunkte werden dann als jeweils einzelne Unterstruktur interpretiert, was natürlich nicht in Übereinstimmung mit dem zugrundeliegenden System steht. Bei diesem Effekt handelt es sich allerdings nicht um ein Problem, das speziell diesen Algorithmus betrifft, sondern um eine generelle Schwierigkeit, die auftritt, sobald man höherdimensionale Räume verwendet.

Um feststellen zu können, ob eine Datenmenge nach einer bestimmten Wahrscheinlichkeitsdichte verteilt ist, gibt es in der Statistik mehrere Verfahren. Das bekannteste davon, das χ^2–Kriterium hat bereits A. Fraser als Abbruchkriterium für seinen Algorithmus mit Erfolg verwendet [7]. Beim χ^2–Kriterium wird der Wahrscheinlichkeitsraum in Abschnitte unterteilt. Die Wahrscheinlichkeit dieser Abschnitte wird mit der erwarteten Wahrscheinlichkeit verglichen. Die ermittelte Abweichung wird mit der χ^2–Verteilung verglichen (aus diesem Grund wird das Kriterium auch χ^2–Kriterium genannt). Die sogenannte *Nullhypothese*, gibt an, ob die zu schätzende Wahrscheinlichkeitsdichteverteilung hinreichend gut mit den erwarteten Werten übereinstimmt. Um diese Entscheidung treffen zu können, muß zunächst ein *Konfidenzniveau* festgelegt werden, das angibt, wie genau die Übereinstimmung sein muß. Das Konfidenzniveau kann aus Tabellen entnommen werden (z.B. [8] oder [10]).

Die algorithmische Umsetzung des χ^2–Kriteriums wird folgendermaßen vorgenommen:

- Zunächst wird der Hyperwürfel in 2^n Unterklassen aufgeteilt (n ist die Anzahl der verwendeten Dimensionen).
- Es wird nun untersucht, zu welchem Subwürfel jeder Datenpunkt zuzuordnen ist. Dabei wird mitgezählt, welcher Subwürfel wieviele Datenpunkte enthält.
- Nun erfolgt die Berechnung von

$$\chi^2 = \sum_{i=1}^{2^n} \frac{(N_i - Np_i)^2}{Np_i}, \tag{3.19}$$

  wobei

  - N die Anzahl der Punkte im übergeordneten Hypervolumen,
  - $p_i = \frac{1}{2^n}$ die Soll–Wahrscheinlichkeit für eine Gleichverteilung und
  - N_i die Anzahl der Punkte in den jeweiligen Unterklassen ist.

- Da der χ^2 Test für große Datenmengen gedacht ist ($N \geq 30$) und bei den vorliegenden Datenstrukturen typisch Werte von etwa $N = 10$ auftreten, ist es notwendig, eine entsprechende Anpassung vorzunehmen. Als günstig hat sich dabei

erwiesen, für die Korrektur die Varianz

$$\sigma_b^2 = \frac{(2^n - 1)^2}{(2^n)^2} \tag{3.20}$$

der entsprechenden Multinomialverteilung zu wählen. Der Vergleich erfolgt daher mit der Größe $\sigma_b^2 \chi^2$.

- Das Resultat $\sigma_b^2 \chi^2$ wird mit dem Konfidenzniveau verglichen. Falls es dieses überschreitet, so wird eine Unterstruktur im betroffenen Hypervolumen angenommen, andernfalls liegt eine Gleichverteilung vor.

  Als Konfidenzniveau hat A. Fraser 20% vorgeschlagen. Die Bedingung, die erfüllt sein muß, damit eine Gleichverteilung postuliert wird, lautet daher

  $$\chi^2 \sigma_b^2 < \chi_{20\%}^2. \tag{3.21}$$

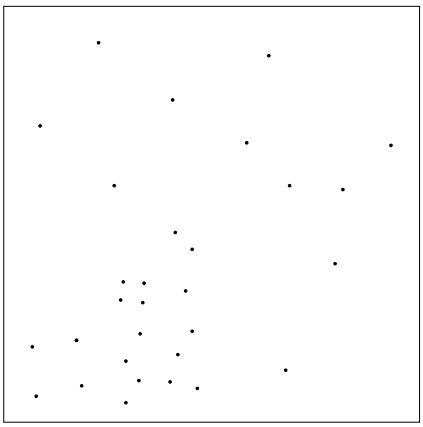
  Bei der Untersuchung höherdimensionaler Verteilungen wurden die Konfidenzniveaus allerdings experimentell angepaßt, da, wie sich bereits bei [2] herausstellte, bei Dimensionen $n \geq 3$ andere Niveaus geeigneter sind, um kleinere Datenmengen zu qualifizieren.

In der Abbildung 3.5 wird ein Beispiel für die rekursive Anwendung des χ^2–Kriteriums auf eine zweidimensionale Wahrscheinlichkeitsdichteverteilung gezeigt. Das erste Teilbild stellt die Verteilung der Datenpunkte im untersuchten Abschnitt dar. Nach der ersten Teilung in $2^2 = 4$ Unterbereiche ergibt sich eine Aufteilung der Datenpunkte auf die Intervalle, wie sie im zweiten Teilbild dargestellt wird. Man erkennt sofort, daß in den einzelnen Bereichen unterschiedlich viele Datenpunkte enthalten sind, es liegt also noch keine Gleichverteilung vor. Daher wird derselbe Test rekursiv für jedes der vier Intervalle durchgeführt.

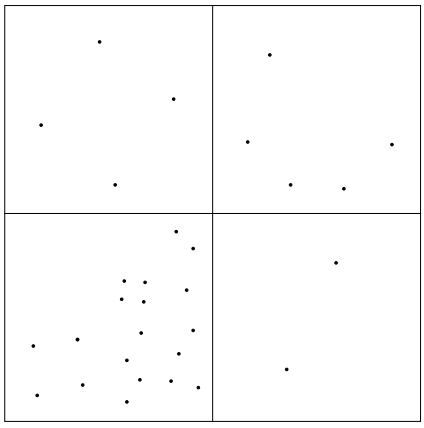
Nach Abschluß der rekursiven Prozedur ergibt sich für den betrachteten Abschnitt eine Aufteilung wie im letzten Teilbild von Abbildung 3.5. In jedem der Teilintervalle liegt nun Gleichverteilung vor (diese Tatsache kann durch erneute Unterteilung jedes Intervalles leicht geprüft werden).

Obwohl dieses Kriterium zunächst sehr grob erscheint, liefert es doch bemerkenswert exakte Ergebnisse. Eine weitere Verbesserung kann man dadurch erreichen, daß die Unterklassen noch einmal unterteilt werden. Nun wird wiederum das χ^2–Kriterium anwendet. Erst wenn alle $2^n + 1$ Tests auf eine Gleichverteilung hinweisen, wird die Unterteilung dann auch tatsächlich abgebrochen. Es hat sich aber in praktischen Versuchen in [1] gezeigt, daß durch dieses Hinzufügen einer weiteren Überprüfung kaum bessere Ergebnisse erreicht werden können, es wird daher in der Implementierung kein zweiter Test mehr durchgeführt.

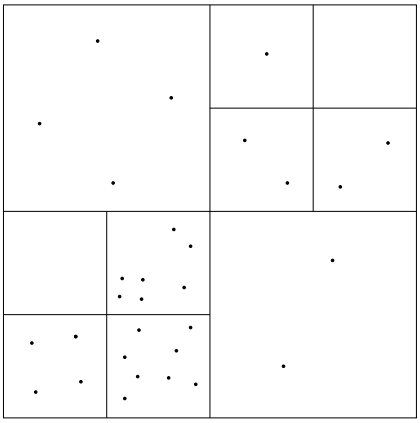
Lediglich bei der ersten Unterteilung (hier wird der gesamte Wahrscheinlichkeitsraum betrachtet) wird eine Ausnahme gemacht. Es wird auf jeden Fall in die Rekursion verzweigt, unabhängig, ob das χ^2–Kriterium erfüllt ist oder nicht. Das ist wie folgt zu begründen: Da die Eingangsdaten oft mittelwertfrei vorliegen, kann es sehr leicht vorkommen, daß bereits nach der ersten Untersuchung in allen Hyperkuben jeweils



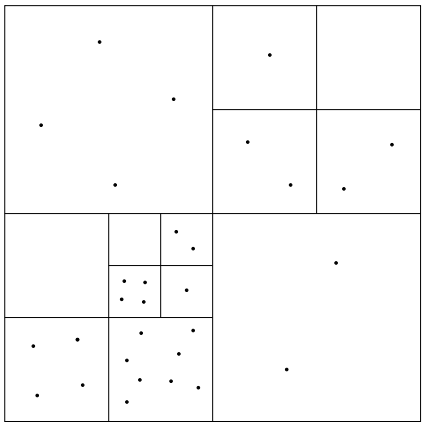
Ausgangsdaten



1. Schritt



2. Schritt



Endergebnis

Abbildung 3.5: Ablauf des χ^2–Tests

annähernd gleich viele Punkte enthalten sind. In diesem Fall würde das χ^2–Kriterium für einen Abbruch der Rekursion sprechen, die resultierende Transinformation wäre 0. Deshalb wird beim ersten Durchlauf auf jeden Fall unterteilt.

Ein weiterer Sonderfall sind Punkte, die sehr eng beieinander liegen (oder im Extremfall sogar gleich sind), während die weitere Umgebung nur spärlich besetzt ist. Sind in einem Intervall ausschließlich Punkte enthalten, auf die diese Beschreibung zutrifft, so werden alle der selben Unterklasse zugeordnet. Das Programm wird daher eine Unterstruktur annehmen, 2^n neue Unterklassen generieren und den Test wiederholen. Durch die räumliche Nähe der Punkte kann es zu beliebig vielen Iterationen kommen, bis die Abbruchbedingung erfüllt ist. Dadurch treten numerische Probleme auf.

Daher ist als Sicherheitsmaßnahme im Programm ein Kriterium vorgesehen, das prüft, ob alle vorhandenen Punkte in eine ϵ–Umgebung passen. Ist dies der Fall, so werden keine neuen Unterklassen erzeugt, sondern die Unterteilung wird beendet, genauso als ob eine Gleichverteilung erkannt worden wäre. Begründet kann diese Maßnahme damit werden, daß bei Trajektorien im Phasenraum Singularitäten auftreten können. Dabei handelt es sich um Artefakte, die durch die Abtastung (zum Beispiel durch eine kommensurable Abtastung einer periodischen Schwingung) entstanden sind. Diese Strukturen sind daher für die Berechnung nicht maßgeblich.

Durch diese Maßnahme wird die Rekursion auch in jenen Klassen beendet, in denen nur ein einziger Datenpunkt enthalten ist.

Bei experimentellen Untersuchungen hat sich ergeben, daß eine geeignete Wahl für ϵ in der Größenordnung von $\frac{1}{100}$ der Klassengröße liegt. Bei größeren Werten kann es passieren, daß vorhandene Strukturen der Wahrscheinlichkeitsdichteverteilung ignoriert werden. Wird ϵ deutlich kleiner gewählt, so wird die Rekursion bei singulären Verteilungsdichten nicht mehr rechtzeitig abgebrochen.

Zusammengefaßt haben adaptive Histogramme folgende Eigenschaften:

- Es wird eine Anpassung der Intervallgröße an die lokale Datenstruktur erreicht.
- Es ist kein Parameter zur Steuerung notwendig. Daher wird beim adaptiven Histogramm kein Vorwissen über die zu analysierenden Daten verlangt.
- In Räumen mit hoher Dimension treten sehr oft große Gebiete auf, die überhaupt keine Datenpunkte enthalten. Während beim Histogramm mit fester Intervallgröße daher zahlreiche leere Intervalle auftreten, die beim Berechnen keinen Beitrag zum Ergebnis liefern, faßt das adaptive Histogramm diese Intervalle zu großen Clustern zusammen. Für das größere Intervall ist nur eine Berechnung erforderlich. Auf diese Weise kann überflüssige Rechenzeit vermieden werden.

Im folgenden wird daher die algorithmische Umsetzung diskutiert und eine Analyse der Ergebnisse vorgenommen.

3.2.1.4 Ergebnisse

Um einen Vergleich mit einem Histogramm mit fester Klassengröße zu ermöglichen, werden die Schätzwerte für die Wahrscheinlichkeitsdichtefunktion, die an sich nur innerhalb des Programmes verwendet werden, ausgewertet. Als Testsignal wird wieder

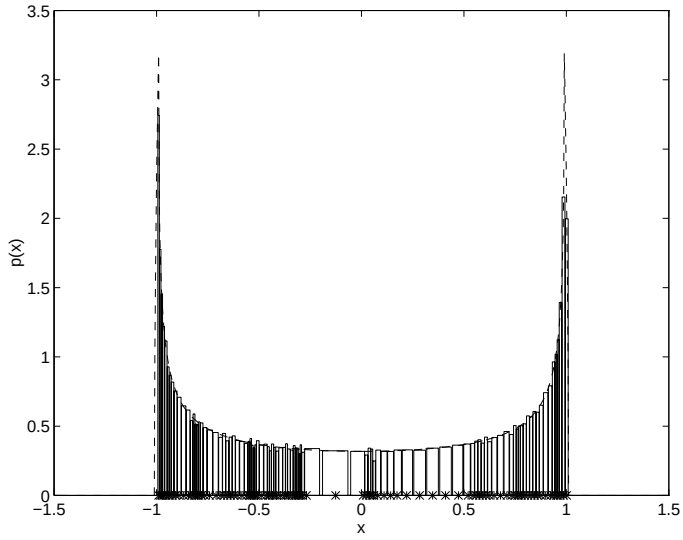


Abbildung 3.6: Adaptives Histogram

die mit 42 dB Signal-Rausch Abstand verrauschte Sinusschwingung verwendet. Das Ergebnis dieser Berechnung ist in Abbildung 3.6 dargestellt. Man erkennt deutlich die unterschiedlichen Intervallgrößen in Bereichen mit flachem bzw. steilem Verlauf der Wahrscheinlichkeitsdichtefunktion. Eine Berechnung des quadratischen Fehlers ϵ^2 nach (3.7) ergibt $\epsilon^2 = 14 \cdot 10^{-3}$. Das bedeutet eine Verbesserung gegenüber dem Histogramm mit fester Klassengröße um über 73%. Für die Erzeugung des adaptiven Histogramms sind außerdem keine Vorkenntnisse über die zu schätzende Funktion nötig.

3.2.2 Berechnung der Transinformation

Es wurde in den vorigen Abschnitten gezeigt, daß ein Algorithmus zur Berechnung der Transinformation zunächst Abschnitte im Phasenraum finden muß, in denen die Wahrscheinlichkeitsdichte konstant ist. Bei der Umsetzung dieses theoretischen Ergebnisses in ein konkretes Programm stellt sich die Frage, wie man die geforderten Intervalle erhält. Strenggenommen müßte die Wahrscheinlichkeitsdichteverteilung bereits bekannt sein, um Abschnitte mit konstanten Funktionswerten wählen zu können.

Als Lösung bietet sich eine rekursive Struktur an, die ausschließlich kubische Unterklassen verwendet. Dabei wird der Signalraum in Richtung jeder Koordinate in zwei gleich große Abschnitte geteilt. Man erhält auf diese Weise 2^n Unterklassen, die alle räumlich disjunkt und gleich groß sind. Beobachtet man in einer der Klassen Gleichverteilung, so kann man, wie bereits gezeigt worden ist, das Teilergebnis berechnen. Sind hingegen noch Unterstrukturen vorhanden, so wissen wir, daß die Intervalle nicht optimal gewählt sind. In diesem Fall wird die Klasse erneut in 2^n Unterklassen aufgeteilt, die jede für sich auf Gleichverteilung hin untersucht werden.

Eine der Einschränkungen, die mit diesem Algorithmus verbunden sind, ist die Festlegung auf orthogonale Hyperwürfel, deren Seitenlänge nur jeweils um einen Faktor 2

variieren kann. Diese Limitierung spielt bei der Transinformation keine Rolle. Falls ein gleichverteilter Bereich existiert, der durch einen Hyperwürfel nicht abgedeckt werden kann, so wird so weit unterteilt, daß sich der Bereich aus mehreren Würfeln zusammensetzt. Für das Ergebnis der Transinformation ist diese Unterteilung belanglos.

Es müssen gleichzeitig mit der Verbundwahrscheinlichkeit auch Unterteilungen für die beiden Randwahrscheinlichkeiten berechnet werden, wobei darauf zu achten ist, daß jedem Bereich der Verbundwahrscheinlichkeit eindeutig ein zugehöriger Bereich in jeder der beiden Randwahrscheinlichkeiten zuzuordnen sein muß. Alleine diese Tatsache erzwingt disjunkte Hyperkuben, da es sonst zu Überlappungen in den Randverteilungen käme, was numerisch kaum zu bewältigen wäre.

Eine Übersicht über den Ablauf der Berechnung geben die Bilder 3.7 und 3.8. In den folgenden Absätzen sind die einzelnen Schritte im Detail erklärt.

Zunächst werden die Vektoren $\vec{x}$ nach (2.1) aus den zeit- und wertdiskret vorliegenden Eingangswerten gebildet und in einer einfach verketteten Liste abgespeichert. Dabei wird der Wertebereich der Vektoren bestimmt. Die ermittelten Begrenzungen werden verwendet, um die Ausmaße des ersten Hyperkubus festzulegen.

Nun wird die Verteilung von $\vec{x}$ bestimmt. Als erstes Intervall wird der gesamte Wertebereich der Eingangsdaten verwendet. Jede Koordinate dieses Intervalles $\vec{x}_i$ wird in zwei gleich große Bereiche unterteilt. Man erhält dadurch im n–dimensionalen Raum 2^n kleinere Hyperkuben, die das ursprüngliche Intervall vollkommen ausfüllen. Die Vektoren werden nun entsprechend ihrer Lage auf die kleineren Hyperkuben aufgeteilt. Mit Hilfe des χ^2–Kriteriums wird festgestellt, ob in $\vec{x}_i$ Gleichverteilung vorliegt, oder ob noch Strukturen in der Wahrscheinlichkeitsdichteverteilung vorhanden sind. An dieser Stelle erfolgt eine Fallunterscheidung:

- Wird Gleichverteilung erkannt, so wird der Wert der Verteilung $p(\vec{x}_i)$ gespeichert. Dieser kann mit
$$p(\vec{x}_i) = \frac{N_i}{N} \tag{3.22}$$
ermittelt werden. N_i ist die Anzahl der im Intervall i enthaltenen Vektoren, N die Gesamtzahl aller Vektoren.

- Wird eine Struktur in der Wahrscheinlichkeitsdichteverteilung erkannt, so wird der Algorithmus für jeden der 2^n kleineren Hyperkuben rekursiv aufgerufen. Damit die Randverteilung in der Berechnung der Transinformation verwendet werden kann, wird jedes dieser Intervalle eindeutig gekennzeichnet.

Die Schätzung der Wahrscheinlichkeitsdichte von $\vec{y}$ erfolgt nach demselben Prinzip. Auch bei der Verbundwahrscheinlichkeitsdichte wird dasselbe Verfahren angewandt. Lediglich nach dem χ^2–Test ergeben sich dort die folgenden Änderungen:

- Liegt Gleichverteilung vor, so wird der Beitrag des Intervalles zur Transinformation aus
$$I(X_i;Y_i) = p(\vec{x}_i,\vec{y}_i)\log\frac{p(\vec{x}_i,\vec{y}_i)}{p(\vec{x}_i)p(\vec{y}_i)} \tag{3.23}$$

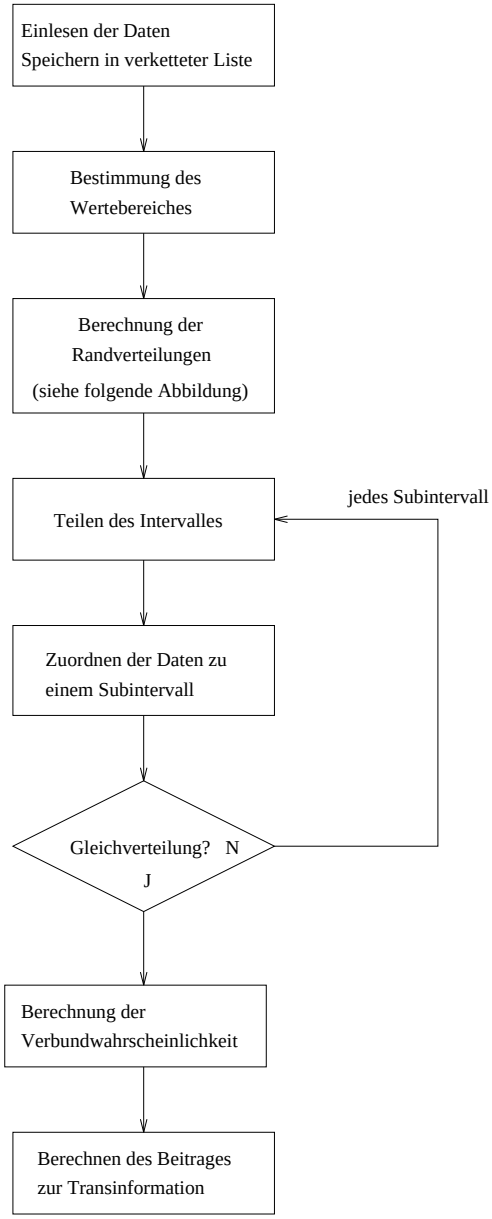


Abbildung 3.7: Ablaufdiagramm zur Berechnung der Transinformation

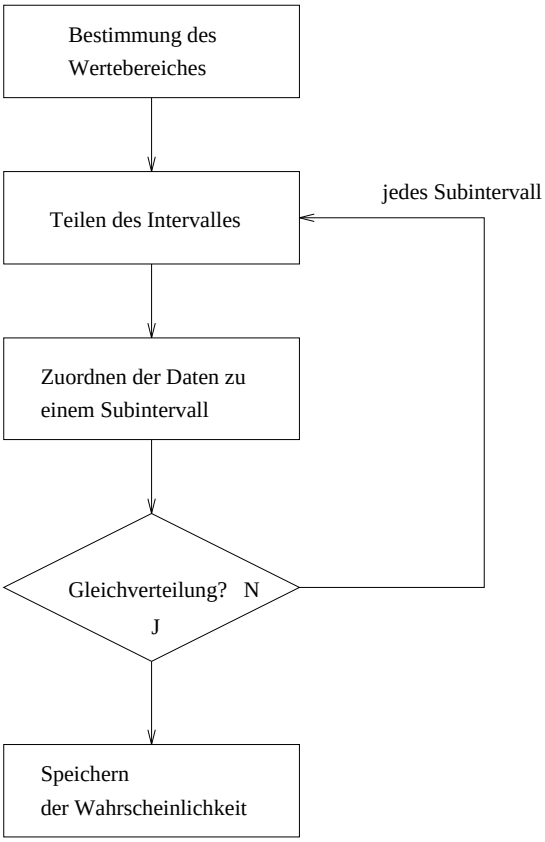


Abbildung 3.8: Ablaufdiagramm zur Ermittlung der Randverteilungen

errechnet. Den Wert der Verbundwahrscheinlichkeitsdichte erhält man aus

$$p(\vec{x}_i, \vec{y}_i) = \frac{N_i}{N}. \tag{3.24}$$

N ist die gesamte Anzahl an Datenvektoren, N_i ist die Zahl derer, die innerhalb des Intervalles i liegen. Die Werte für die Wahrscheinlichkeiten der Randverteilungen werden aus den zuvor ermittelten Wahrscheinlichkeiten $p(\vec{x}_i)$ und $p(\vec{y}_i)$ bestimmt. Die Tatsache, daß in der Rekursion die Teilungen immer in 2^n Hyperkuben erfolgen, ermöglicht es, zu jedem Intervall der Verbundwahrscheinlichkeitsdichte einen entsprechenden Hyperkubus in der Randverteilung zuzuordnen.

Die Bearbeitung dieses Intervalles ist damit abgeschlossen, das Ergebnis wird zum Gesamtergebnis addiert, das heißt an die darüberliegende Stufe der Rekursion weitergeleitet.

- Falls man eine Struktur innerhalb des Intervalles erkennt, wird der Vorgang für jeden darin enthaltenen Hyperkubus rekursiv wiederholt. Die Transinformation $I(X_i; Y_i)$ für das Intervall errechnet sich abschließend aus

$$I(X_i; Y_i) = \sum_{j=1}^{2^n} I(X_{ij}; Y_{ij}), \tag{3.25}$$

wobei $I(X_{ij}; Y_{ij})$ die Transinformation des j–ten Hyperkubus ist, die aus der Rekursion erhalten wurde. Das Ergebnis wird an die aufrufende Funktion übergeben.

Ist die Rekursion abgeschlossen, erhält man den Schätzwert für die Transinformation $I(X;Y)$. Die Berechnung ist damit abgeschlossen.

3.2.3 Berechnung der Kullback Leibler Distanz

Die Berechnung der Kullback Leibler Distanz kann analog zur Berechnung der Transinformation durchgeführt werden. Da zur Ermittlung der Kullback Leibler Distanz aber die Verbundwahrscheinlichkeitsdichte nicht benötigt wird, müssen nur die zwei Randverteilungsdichten geschätzt werden.

Eine Übersicht über den Ablauf der Berechnung ist im Bild 3.9 dargestellt. In den folgenden Absätzen werden die einzelnen Schritte im Detail erklärt.

Aus den skalaren Eingangsdaten wird der Mittelwert und die mittlere Leistung der beiden Eingangssignale errechnet. Bei der Bildung der Vektoren aus den Eingangsdaten werden diese auf Mittelwert $\mu = 0$ und mittlere Leistung $\bar{P} = 1$ normiert. Dies ist notwendig, da bei der Berechnung der Kullback Leibler Distanz nach (2.21) die Wahrscheinlichkeitsdichte $p(\vec{x})$ durch $q(\vec{x})$ dividiert wird. Existiert ein einziger Bereich $\vec{x_i}$ in dem $p(\vec{x_i}) \neq 0$ und $q(\vec{x_i}) = 0$ ist, so wird die Kullback Leibler Distanz

$$D(p||q) = \infty, \tag{3.26}$$

was eine Interpretation unmöglich macht. Daher müssen die beiden Signale den gleichen Wertebereich aufweisen, um ein sinnvolles Ergebnis zu liefern.

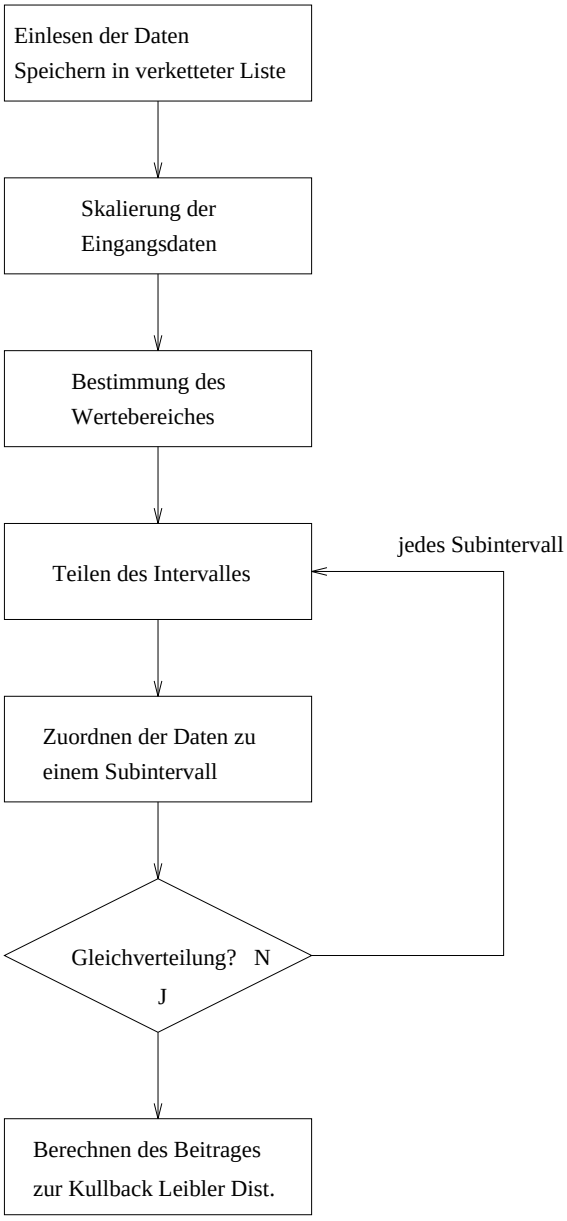


Abbildung 3.9: Ablaufdiagramm zur Ermittlung der Kullback Leibler Distanz

Wie bei der Berechnung der Transinformation wird der gesamte Wertebereich als erstes Intervall verwendet. Der rekursive Algorithmus läuft folgendermaßen ab:

Jede Koordinate des Intervalles $\vec{x}_i$ wird in zwei gleich große Teile geteilt. Man erhält dadurch 2^n kleinere Hyperkuben. Die Vektoren von $p(\vec{x}_i)$ und $q(\vec{x}_i)$ werden nun dem entsprechenden Hyperkubus zugeordnet. Nachdem diese Aufteilung beendet ist, wird für $p(\vec{x}_i)$ und $q(\vec{x}_i)$ getrennt der χ^2–Test durchgeführt. In Abhängigkeit von den Ergebnissen dieser beiden Tests wird eine der folgenden Aktionen ausgeführt:

- Falls für beide Wahrscheinlichkeitsverteilungen noch Unterstrukturen vermutet werden, wird derselbe Algorithmus rekursiv für jeden der 2^n Unterbereiche aufgerufen. Das Ergebnis $D\left(p\left(\vec{x}_i\right)||q\left(\vec{x}_i\right)\right)$ für das Intervall i wird aus den Teilergebnissen der Rekursion mit

$$D\left(p\left(\vec{x}_i\right)||q\left(\vec{x}_i\right)\right) = \sum_{j=1}^{2^n} D\left(p\left(\vec{x}_{ij}\right)||q\left(\vec{x}_{ij}\right)\right) \tag{3.27}$$

  ermittelt. $\vec{x}_{ij}$ ist dabei der Hyperkubus j, der im Intervall i enthalten ist.

  Dieses Verfahren wird *nicht* angewandt, wenn einer der Unterbereiche $q(\vec{x}_{ij})$ von $q(\vec{x}_i)$ keine Vektoren enthält. Da diese Wahrscheinlichkeitsdichte bei der Berechnung der Kullback Leibler Distanz im Nenner vorkommt, wäre das Ergebnis unendlich groß. Um solche Singularitäten zu vermeiden, wird in so einem Fall angenommen, daß bereits in $q(\vec{x}_i)$ Gleichverteilung vorliegt, unabhängig vom Ergebnis des χ^2–Tests.

- Liegt für beide Verteilungen Gleichverteilung vor, so kann der Beitrag dieses Intervalls zur Kullback Leibler Distanz mit

$$D\left(p\left(\vec{x}_i\right)||q\left(\vec{x}_i\right)\right) = p\left(\vec{x}_i\right)\log\frac{p\left(\vec{x}_i\right)}{q\left(\vec{x}_i\right)} \tag{3.28}$$

  ermittelt werden. Die Wahrscheinlichkeiten $p(\vec{x}_i)$ und $q(\vec{x}_i)$ errechnen sich aus

$$p(\vec{x}_i) = \frac{N_{pi}}{N_p} \tag{3.29}$$

  und

$$q(\vec{x}_i) = \frac{N_{qi}}{N_q}. \tag{3.30}$$

  N_{pi} und N_{qi} sind die Anzahl der Vektoren des Signals P bzw. Q im Intervall i, N_p und N_q stehen für die Anzahl der insgesamt in diesen Signalen vorhanden Vektoren. Das Ergebnis wird an die darüberliegende Stufe der Rekursion übergeben und die Rekursion beendet.

- Liegt bei einer der beiden Eingangsgrößen Gleichverteilung vor, während in der anderen noch Strukturen vorhanden sind, so wird zunächst die Wahrscheinlichkeitsdichte des gleichverteilten Intervalles mit der Beziehung

$$q(\vec{x}_i) = \frac{N_{qi}}{N_q} \tag{3.31}$$

Datenpunkte	n = 1	n = 2	n = 3
4k	5,01	6,88	6,57
8k	5,04	6,99	6,71
16k	5,02	7,05	6,93
32k	5,04	7,10	7,05
64k	5,00	7,01	6,99
128k	4,90	6,99	7,07
256k	4,94	6,96	7,08
512k	4,93	6,97	7,08

Tabelle 3.1: Transinformation in Bit einer mit SNR = 42 dB verrauschten Sinusschwingung, berechnet mit adaptiven Histogrammen

berechnet, falls es sich um das Signal Q handelt, für das Signal P gilt die äquivalente Beziehung.

Nun wird, wie im ersten Punkt, der Algorithmus rekursiv aufgerufen, mit dem Unterschied, daß nur noch für eines der beiden Signale eine Unterteilung mit anschließendem χ^2–Test durchgeführt wird, während die bereits berechnete Wahrscheinlichkeitsdichte nur noch als Konstante übergeben wird.

Die Kullback Leibler Distanz dieses Intervalles wird aus den Ergebnissen der Rekursion durch

$$D\left(p\left(\vec{x}_i\right)||q\left(\vec{x}_i\right)\right) = \sum_{j=1}^{2^n} D\left(p\left(\vec{x}_i\right)||q\left(\vec{x}_{ij}\right)\right), \tag{3.32}$$

falls $p(\vec{x}_i)$ gleichverteilt ist, andernfalls durch

$$D\left(p\left(\vec{x}_i\right)||q\left(\vec{x}_i\right)\right) = \sum_{j=1}^{2^n} D\left(p\left(\vec{x}_{ij}\right)||q\left(\vec{x}_i\right)\right) \tag{3.33}$$

berechnet und an die aufrufende Funktion übergeben.

3.2.4 Ergebnisse

Bei einer Sinusschwingung handelt es sich um ein deterministisches Signal. Ist zu einem beliebigen Zeitpunkt der Systemzustand vollständig bekannt, so kann der Funktionswert für alle anderen Zeitpunkte exakt berechnet werden. Bei den bisher gebrachten Beispielen wurde jedoch ein mit 42 dB verrauschtes Sinussignal verwendet. Selbst bei exakter Kenntnis des Systemzustandes können nun Funktionswerte nur noch mit einer gewissen Unsicherheit vorhergesagt werden. Die Transinformation beträgt daher 42 dB, das sind 7 Bit.

In Tabelle 3.1 sind Ergebnisse zusammengestellt, die mit dem entwickelten Algorithmus für dieses Sinussignal erhalten werden. Bei den Berechnungen wurde die Anzahl der

Ausgangsdaten *Kerndichte Schätzung*

Abbildung 3.10: Prinzip der Kerndichte Schätzung

Datenpunkte und die bei der Konstruktion des Phasenraumes verwendete Dimension variiert. Es zeigt sich, daß die Ergebnisse sehr knapp am theoretisch ermittelten Wert liegen. Ausgenommen davon ist die Rekonstruktion in einer einzigen Dimension. Da in diesem Fall keine Information über den zeitlichen Zusammenhang vorliegt (bei einer Dimension wird keine Zeitverzögerung eingesetzt), kann auch keine exakte Prädiktion erfolgen. Die in den anderen Spalten vorhandenen kleineren Ungenauigkeiten und Schwankungen haben folgende Ursachen:

- Die statistische Natur der Eingangsdaten. Besonders bei den Berechnungen mit nur wenigen Datenpunkten kann das überlagerte Rauschen vom Algorithmus nicht vollkommen ausgemittelt werden. Auch die Lage der Abtastzeitpunkte spielt hier eine Rolle. Man kann in der Tabelle 3.1 erkennen, daß die Abweichung vom exakten Ergebnis kleiner wird, je mehr Datenpunkte für die Berechnung verwendet werden.

- Die Wahl des Konfidenzniveaus im χ^2–Test. Bei der verwendeten Schwelle von 20% ist zwar eine gute Unterscheidung zwischen gleichverteilten Bereichen und solchen, die noch Feinstrukturen enthalten gewährleistet, es kann aber in Einzelfällen noch immer zu Fehlentscheidungen kommen. Abhilfe könnte zum Beispiel durch die Wahl eines niedrigeren Konfidenzniveaus geschaffen werden, dafür müßte die Zahl der notwendigen Datenpunkte allerdings extrem gesteigert werden, was selbst bei gleicher Genauigkeit die Rechenzeit deutlich steigern würde.

3.3 Berechnung durch Kerndichte Schätzer

3.3.1 Schätzung der Wahrscheinlichkeitsdichteverteilung

3.3.1.1 Prinzip

Ein weiteres Verfahren zur Schätzung von unbekannten Wahrscheinlichkeitsdichtefunktionen ist die Kerndichte Schätzung. Bei dieser Methode wird über jeden Datenpunkt eine Funktion (der sogenannte Kern) gelegt. Die Summe dieser Funktionen bildet dann die geschätzte Wahrscheinlichkeitsdichte $\hat{f}(x)$. In Abbildung 3.10 wird das

Prinzip dieses Vorgangs dargestellt. Mathematisch formuliert bedeutet das im eindimensionalen Fall:

$$\hat{f}(x) = \frac{1}{Nh} \sum_{i=1}^{N} K(\frac{x - x_i}{h}). \tag{3.34}$$

Dabei ist $K(s)$ die Kernfunktion, N die Anzahl der Datenpunkte, x_i sind die Eingangsdaten und h ist ein Parameter, der oft als Kernbreite oder Glättungsparameter bezeichnet wird. Mit h kann eingestellt werden, wie weit der Einfluß eines Datenpunktes reicht. Da der Beitrag eines Datenpunktes zur Wahrscheinlichkeitsdichte immer $1/N$ beträgt, muß das Integral über die Kernfunktion immer

$$\int_{-\infty}^{\infty} K(s)\, ds = 1 \tag{3.35}$$

ergeben, damit auch das Integral

$$\int_{-\infty}^{\infty} \hat{f}(x)\, dx = 1 \tag{3.36}$$

ergibt.

Im n–dimensionalen Raum läßt sich $\hat{f}(x)$ durch

$$\hat{f}(\vec{x}) = \frac{1}{Nh_1 \ldots h_n} \sum_{i=1}^{N} \left\{ \prod_{j=1}^{n} K\left(\frac{x_j - x_{ij}}{h_j} \right) \right\} \tag{3.37}$$

berechnen. Es kann für jede Koordinate eine eigene Kernbreite gewählt werden. Häufig werden die Kernbreiten $h_1 = \ldots = h_n = h$ aber identisch gewählt. Mit x_j wird die Komponente j des Vektors $\vec{x}$ bezeichnet, x_{ij} bezeichnet die Komponente j des Vektors $\vec{x}_i$.

Je nach Wahl der Kernfunktion kann man völlig unterschiedliche Resultate bei der geschätzten Verteilungsfunktion erreichen. Erzeugt man beispielsweise Kerne mit der Dirac–Funktion $\delta(x)$ oder der Sprungfunktion $\sigma(x)$ so kann man mit

$$K(s) = \delta(s), \tag{3.38}$$

ein Streudiagramm[1] erhalten, mit

$$K(s) = \sigma(s + 0.5) - \sigma(s - 0.5) \tag{3.39}$$

erhält man ein gemitteltes, verschobenes Histogramm (siehe [12]). Tatsächlich wird in [13] gezeigt, daß sich alle nichtparametrischen Schätzverfahren durch Kerndichte Schätzer beschreiben lassen. Nichtparametrische Schätzverfahren sind solche, bei denen das Ergebnis als diskrete Menge geschätzter Werte vorliegt, im Gegensatz zu den parametrischen Schätzverfahren. Bei diesen wird eine analytische Schätzfunktion durch einen oder mehrere zu wählende Parameter an die vorliegenden Daten angepaßt.

[1] Ein Streudiagramm erhält man, wenn man für jedem Datenpunkt eine Markierung an die entsprechende Stelle setzt. Streudiagramme werden besonders gern zur schnellen Beurteilung von zweidimensionalen Verteilungsdichten eingesetzt.

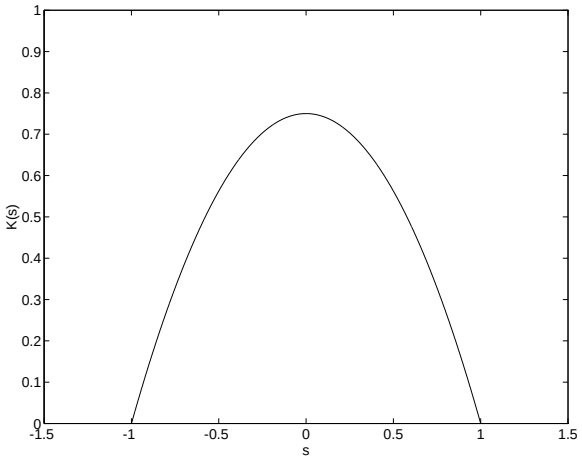


Abbildung 3.11: Epanechnikov Kern

3.3.1.2 Wahl der Kernfunktion

V. Epanechnikov konnte 1969 zeigen, daß im Sinne eines asymptotischen gemittelten integrierten quadratischen Fehlers [6] die optimale Kernfunktion durch

$$K(s) = \frac{3}{4}(1 - s^2), \quad |s| < 1 \tag{3.40}$$

beschrieben wird (siehe auch Bild 3.11).

Es wird aber gleichzeitig gezeigt, daß die Wahl der Kernfunktion keinen sehr großen Einfluß auf die Genauigkeit der resultierenden Schätzung hat. Der Kern

$$K(s) = 1 - |s|, \quad |s| < 1 \tag{3.41}$$

beispielsweise verursacht zwar einen größeren mittleren Fehler, dafür ist der Rechenaufwand geringer als beim optimalen Kern. Die in der Statistik häufig verwendete Gaußfunktion

$$K(s) = \frac{1}{\sqrt{2\pi}} e^{-\frac{|s|^2}{2}} = N(0,1) \tag{3.42}$$

weist ebenfalls einen größeren mittleren Fehler als der Epanechnikov Kern auf, auch der Rechenaufwand ist höher. Dafür ist die geschätzte Wahrscheinlichkeitsdichtefunktion im Gegensatz zu den bisher vorgestellten Kernfunktionen stetig differentierbar, was bei bestimmten Anwendungen notwendig ist.

Da sich die Wahl der Kernfunktion also kaum auf das Ergebnis auswirkt, kann man sich für eine Funktion entscheiden, die andere Vorzüge hat als eine etwas höhere Effizienz. Diese sind, wie bereits erwähnt, beispielsweise einfache Berechenbarkeit, Stetigkeit oder Differenzierbarkeit. Je nach der Aufgabenstellung kann einer bestimmten Größe dabei mehr Bedeutung zugemessen werden.

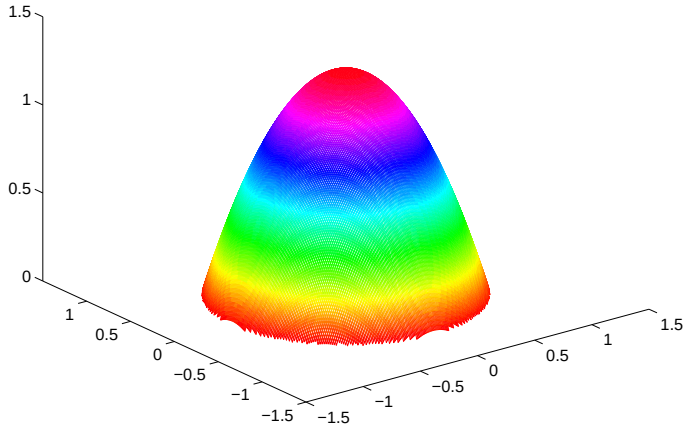


Abbildung 3.12: Epanechnikov Kern für $n = 2$

Die bisher vorgestellten Schätzfunktionen haben gemeinsam, daß sie ausschließlich positive Werte liefern. Bei der graphischen Darstellung von Datenmengen gibt es Fälle, bei denen man mit Funktionen höherer Ordnung, die auch negative Anteile enthalten, gute Ergebnisse erzielen kann. Bei diesen kommt es vor allem auf eine gute Wiedergabe des optischen Eindruckes der Verteilungsfunktion an, wobei die absolute Genauigkeit an einzelnen Punkten nur ein zweitrangiges Kriterium ist.

Die Existenz von Kernfunktionen mit negativen Funktionswerten wird an dieser Stelle nur erwähnt. Solche Funktionen sind unbrauchbar für das vorliegende Problem, da negative Wahrscheinlichkeiten auftreten können.

Die bisher vorgestellten Kernfunktionen wurden alle für den eindimensionalen Fall angegeben. Eine Erweiterung auf höherdimensionale Funktionen ist unproblematisch, es sind allerdings einige Festlegungen nötig. Im Prinzip kann die Einhüllende eines mehrdimensionalen Kerndichte Schätzers in jeder Raumrichtung einen anderen Verlauf aufweisen. Im Rahmen dieser Arbeit beschränken wir uns auf isotrope Kernfunktionen, da bei den vorliegenden Problemen alle Richtungen im Phasenraum gleichwertig sind. Dadurch können wir auch auf eine Notation $K(\vec{s})$ verzichten und weiterhin $K(s)$ anschreiben, wobei für das Argument der Betrag $|\vec{s}|$ eingesetzt wird.

Da die Kernfunktion natürlich auch im n–dimensionalen Fall die Bedingung

$$\oint_{\mathbb{R}^n} K\left(|\vec{s}|\right) d\vec{s} = 1 \tag{3.43}$$

erfüllen muß, ist ein Vorfaktor nötig. Mathematisch korrekt müßte man deswegen $K_n\left(|\vec{s}|\right)$ schreiben, da die Skalierung der Kernfunktion von der Dimension abhängt. Auf Grund der einfacheren Lesbarkeit wird im Weiteren angenommen, daß die Skalierung der Kernfunktion jeweils der Dimension des Argumentes entspricht.

Beispiele für n–dimensionale Kernfunktionen sind beispielsweise der Gauß Kern

$$K(\vec{s}) = \frac{1}{(2\pi)^{n/2}} e^{-\frac{|\vec{s}|^2}{2}} \tag{3.44}$$

und der in Bild 3.12 dargestellte Epanechnikov Kern

$$K(\vec{s}) = \begin{cases} \frac{n+2}{2V(n)} \left(1 - |\vec{s}|^2\right) & : \quad |\vec{s}| < 1 \\ 0 & : \quad sonst, \end{cases} \tag{3.45}$$

wobei $V(n)$ dem Volumen einer n–dimensionalen Hyperkugel entspricht, also

$$V(n) = \frac{1}{2}\pi^{n/2}\Gamma(n/2+1). \tag{3.46}$$

Ein wesentlicher Parameter bei der Kerndichte Schätzung ist die Kernbreite h. Wählt man sie zu groß, so werden Detailstrukturen von den Kernfunktionen überdeckt. In diesem Fall geht Information verloren. Wird dagegen die Kernbreite zu schmal gewählt, erhält man eine Schätzung für die Wahrscheinlichkeitsdichtefunktion, in der Detailstrukturen enthalten sind, die in der zu schätzenden Funktion nicht vorkommen. Im Kapitel 3.3.3 wird näher auf diese Problematik eingegangen.

Für die Schätzung der Wahrscheinlichkeitsdichteverteilung sind die folgenden Eigenschaften wesentlich:

- Durch das Überlagern von kontinuierlichen Funktionen über die einzelnen Datenpunkte erhält man auch bei (im Vergleich zum Histogramm) wenigen Datenpunkten bereits eine glatte Summenverteilung in (3.34).
- Es wird ein Parameter, nämlich die Kernbreite verwendet. Da die Wahl der optimalen Kernbreite von der Struktur der Eingangsdaten abhängt, werden, ähnlich wie beim Histogramm, Vorkenntnisse über die zu schätzende Funktion benötigt.

Die angeführten Eigenschaften lassen vermuten, daß Kerndichte Schätzer zwar prinzipiell gut für die gegebene Problemstellung geeignet sind, jedoch eventuell bei der praktischen Verwendung Schwierigkeiten verursachen. Im weiteren Verlauf dieser Arbeit werden zwei verschiedene Verfahren beschrieben, ein konventioneller Lösungsansatz sowie ein von Matt Kennel entworfener, vereinfachter Algorithmus.

3.3.2 Berechnung der Transinformation

In die Formel (2.30) für die Berechnung der Transinformation wird für die Verbundwahrscheinlichkeit der aus den überlagerten Kernfunktionen gebildete Schätzwert eingesetzt.

Da es sich bei den Kernfunktionen um kontinuierliche Funktionen handelt, muß, bevor die Summe berechnet werden kann, eine Diskretisierung vorgenommen werden. Dazu werden im Wertebereich Stützstellen definiert, an denen die geschätzte Wahrscheinlichkeitsdichtefunktion abgetastet wird.

Die Summation erfolgt im gesamten Bereich, in dem die Schätzfunktion $\hat{f}(x) \neq 0$ ist. Bei Kernfunktionen mit unendlicher Ausdehnung (beispielsweise dem Gaußkern) muß ein daher Schwellwert definiert werden, ab dem der Beitrag zur Transinformation vernachlässigbar ist. Kerne mit beschränkter Ausdehnung (beispielsweise der Epanechnikov Kern) lassen eine vollständige Summation zu.

Bei der Summation über $\hat{f}(x)$ handelt es sich um eine numerische Integration mit fixen Stützstellen. Darin liegt bereits der wesentlichste Unterschied zur Histogrammethode. Während Histogramme bei zu feiner Aufteilung der Intervalle keine brauchbaren Ergebnisse bringen (siehe 3.1.2), steigt die Rechengenauigkeit bei der Verwendung von Kerndichte Schätzern mit der Zahl der Summationspunkte. Der Grund dafür ist, daß bei Kerndichte Schätzern auch zwischen zwei benachbarten Datenpunkten noch Funktionswerte für $\hat{f}(x)$ definiert sind, während bei der Verwendung eines Histogrammes in diesem Fall ein leeres Intervall auftreten würde.

Um ein hinreichend genaues Ergebnis zu erhalten, muß sichergestellt werden, daß jede Kernfunktion in jeder Koordinate mehrmals abgetastet wird. Für den Epanechnikov- und den Gauß–Kern haben experimentelle Untersuchungen ergeben, daß ab etwa acht Abtastpunkten pro Kern der durch die numerische Integration verursachte Fehler unter 1% bleibt. Die Eingangsdaten sind normiert, ihr Wertebereich ist daher in etwa das Intervall $[-1 \quad 1]$. Bei einer Kernbreite von $h = 0,1$ werden daher ungefähr $\frac{1+1}{0,1/8} = 160$ Summationspunkte für jede Koordinate benötigt, um einen sinnvollen Schätzwert zu erhalten. Dies ist, wie im vorigen Kapitel gezeigt wurde, eine sehr optimistische Schätzung, da in der Praxis eine Kernbreite $h = 0,1$ meistens zu grob gewählt ist.

Während die Rechenungenauigkeit mit kleineren Intervallen sinkt, steigt andererseits dadurch die benötigte Rechenzeit. Werden für N Eingangsdaten im n–dimensionalen Raum für jede Koordinate K Stützpunkte berechnet, so muß insgesamt K^n Mal die Summe aus N Kernfunktionen gebildet werden. Der Rechenaufwand wird also besonders bei höheren Dimensionen ein Kriterium für die Verwendbarkeit dieses Algorithmus sein (siehe 4.5.3).

Eine Übersicht über den Ablauf der Berechnung ist im Bild 3.13 dargestellt. Im den folgenden Absätzen werden die einzelnen Schritte im Detail erklärt.

Aus den skalaren Eingangsdaten wird der Mittelwert und die mittlere Leistung der beiden Eingangssignale errechnet. Bei der Bildung der Vektoren aus den Eingangsdaten werden diese auf Mittelwert $\mu = 0$ und mittlere Leistung $\bar{P} = 1$ normiert. Durch diese Skalierung können die Kernbreiten unabhängig von den Eingangssignalen in absoluten Zahlenwerten angegeben werden. Die Vektoren werden in einer einfach verketteten Liste gespeichert.

Nun werden nacheinander die geschätzten Werte $\hat{p}(\vec{x})$, $\hat{p}(\vec{y})$ und $\hat{p}(\vec{x},\vec{y})$ für die Wahrscheinlichkeitsdichteverteilungen $p(\vec{x})$, $p(\vec{y})$ und $p(\vec{x},\vec{y})$ berechnet. Dazu wird der n–dimensionale Raum an K^n Stützstellen gleichmäßig abgetastet, also K Mal je Koordinate. Am Punkt $\vec{k}_i$ wird die Summe

$$\hat{p}(\vec{k}_i) = \frac{1}{N}\frac{1}{h}\sum_{j=1}^{K} K\left(\frac{\left|\vec{k}_i - \vec{s}_j\right|}{h}\right) \tag{3.47}$$

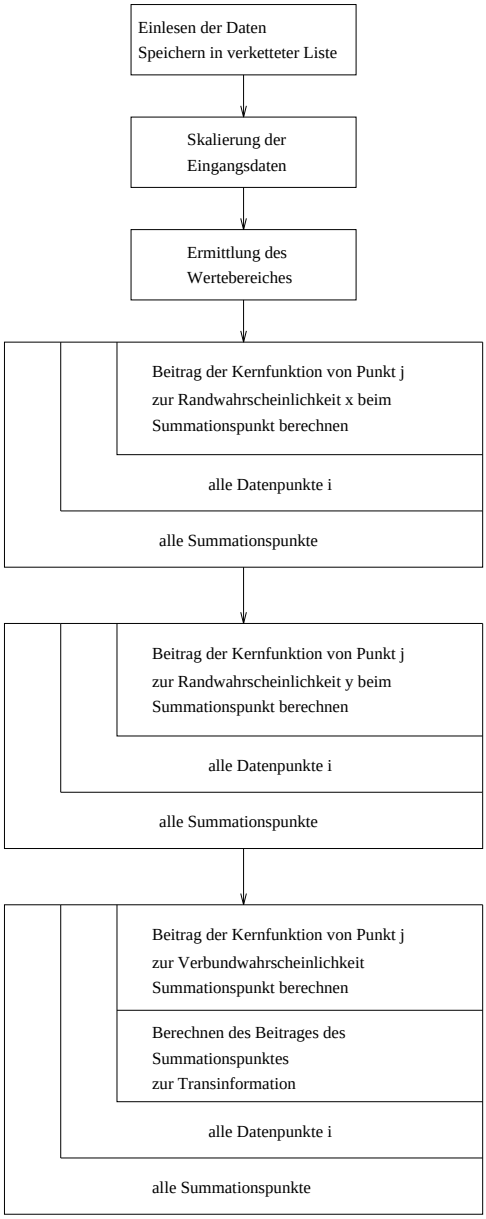


Abbildung 3.13: Ablaufdiagramm der Berechnung durch Kerndichte Schätzer

gebildet. $\vec{s}_j$ ist dabei der j-te Datenpunkt für die zu berechnende Wahrscheinlichkeitsdichteverteilung.

Für die Randverteilungen werden die Ergebnisse im Speicher abgelegt, bei der Ermittlung der Verbundwahrscheinlichkeitsdichte werden sie dazu verwendet, um den Beitrag $I_i(X;Y)$ der Stützstelle zur Transinformation $I(X;Y)$ zu ermitteln. $I_i(X;Y)$ wird durch die Beziehung

$$I_i(X;Y) = \hat{p}(\vec{x}_i, \vec{y}_i) \log \frac{\hat{p}(\vec{x}_i, \vec{y}_i)}{\hat{p}(\vec{x}_i)\hat{p}(\vec{y}_i)}. \tag{3.48}$$

beschrieben.

Die Transinformation erhält man als Ergebnis der Summation über alle berechneten Schätzwerte, also

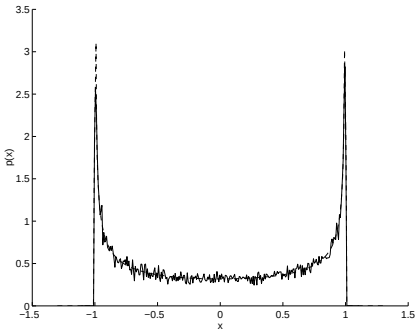
$$I(X;Y) = \sum_{i=1}^{K^n} I_i(X;Y). \tag{3.49}$$

3.3.3 Ergebnisse

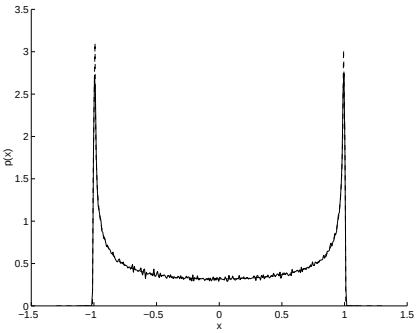
Durch eine Modifikation des Programmes werden die Schätzwerte für die Randverteilungen, die vom Programm intern verwendet werden, ausgegeben. Dadurch ist es möglich, die Schätzung der Wahrscheinlichkeitsdichteverteilung durch den Algorithmus zu überprüfen. In der Abbildung 3.14 sind die geschätzten eindimensionalen Randverteilungsdichten für verschiedene Kernbreiten h und 300 Stützpunkte abgebildet. Die Sinusschwingung ist für die Wahl einer geeigneten Kernbreite ein besonders unangenehmer Fall, da ihre Wahrscheinlichkeitsdichteverteilung zwei Singularitäten aufweist. Die Steilheit der Flanken wird nur durch die Amplitude des Rauschens begrenzt.

Wählt man die Kernbreite größer als die Rauschamplitude, so wird die geschätzte Wahrscheinlichkeitsdichte verschmiert, wird die Kernbreite zu klein gewählt, so treten lokale Unregelmäßigkeiten auf. Der Grenzfall einer Kernbreite von Null wäre identisch mit einem Histogramm mit unendlich schmalen Intervallen. Es treten daher in diesem Fall die selben Probleme auf, die beim Histogramm diskutiert wurden. In beiden Fällen steigt der Schätzfehler deutlich an. In der Tabelle 3.2 sind die quadratischen Fehler für verschiedene Kernbreiten angeführt. Man erkennt, daß sich Kernbreiten von etwa $h \approx 0,01$ am besten zur Schätzung dieser Wahrscheinlichkeitsdichteverteilung eignen.

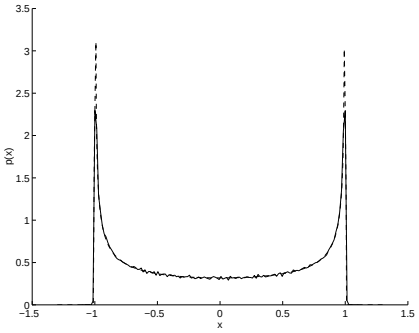
Im Gegensatz zur Kernbreite gibt es bei der Anzahl der Stützpunkte keinen optimalen Wert. Je mehr Stützstellen verwendet werden, desto genauer wird das Ergebnis. Die Tabellen 3.3 und 3.4 zeigen für die Kernbreiten $h = 0,001$ und $h = 0,1$ den quadratischen Fehler bei unterschiedlich vielen Stützpunkten. Es zeigt sich, daß ab einer gewissen Anzahl an Stützpunkten kaum mehr eine Verbesserung des Fehlers erzielt werden kann. Ab diesem Schwellwert sind die einzelnen Kernfunktionen hinreichend gut abgetastet. Die Schwelle hängt daher unmittelbar mit der verwendeten Kernbreite zusammen. Während beispielsweise für eine Kernbreite von $h = 0,01$ der Schwellwert etwa bei 350 Stützstellen erreicht ist, benötigt man bei $h = 0,001$ schon über 500 Stützpunkte. Im Gegensatz zur Kernbreite wirkt sich die Anzahl der verwendeten Stützpunkte direkt auf die Rechenzeit aus (siehe auch Kapitel 4.5.3). Es ist daher ein Kompromiß zwischen Rechengenauigkeit und Ausführungsgeschwindigkeit erforderlich.



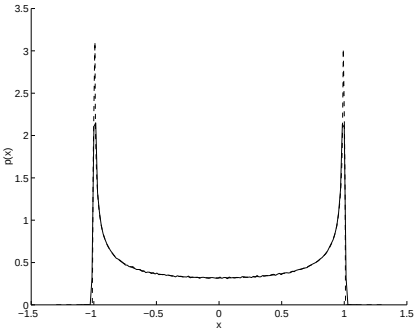
$h = 0,001$



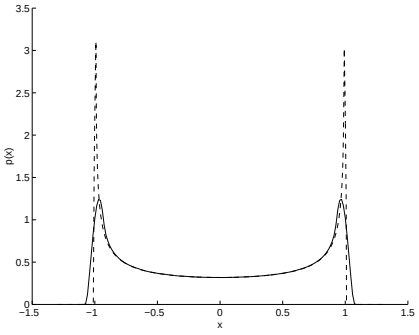
$h = 0,005$



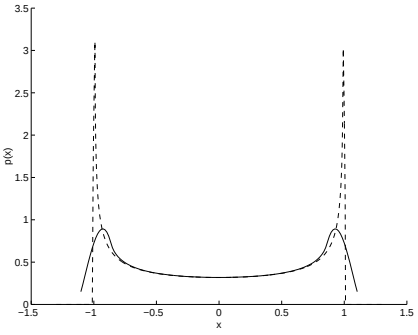
$h = 0,01$



$h = 0,02$



$h = 0,1$



$h = 0,2$

Abbildung 3.14: Kerndichte Schätzung einer verrauschten Sinusschwingung

Kernbreite	Fehler
0,001	54,8
0,005	43,7
0,010	40,7
0,020	41,7
0,050	118,8
0,100	239,8
0,200	378,2

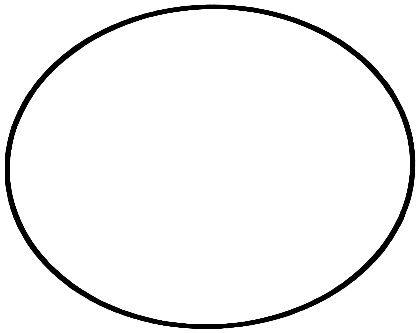
Tabelle 3.2: Quadratischer Fehler bei der Kerndichteschätzung einer Sinusschwingung mit SNR = 42 dB

Stützpunkte	Fehler
150	40,7
200	24,5
250	23,4
300	16,9
350	10,1
400	10,4
450	10,3

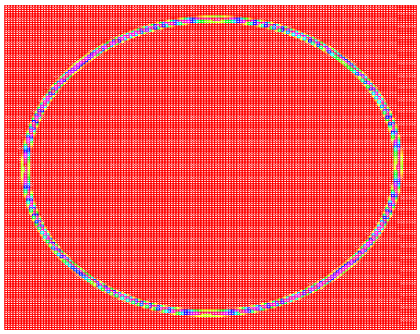
Tabelle 3.3: Quadratischer Fehler für $h = 0,01$

Stützpunkte	Fehler
150	54,8
200	33,9
250	47,0
300	31,7
350	20,7
400	18,2
450	21,3
500	19,6
550	14,6
600	18,2
650	16,7
700	14,4

Tabelle 3.4: Quadratischer Fehler für $h = 0,001$



Originaldaten



Kerndichte Schätzung

Abbildung 3.15: Verrauschte Sinusschwingung in zwei Dimensionen

Abbildung 3.15 zeigt die geschätzte Wahrscheinlichkeitsdichte für zwei Dimensionen. Als Verzögerungszeit T wurde 1/4 Periodendauer gewählt. Im linken Bild sind die Originaldaten eingetragen, das rechte Bild gibt die geschätzte Wahrscheinlichkeitsdichteverteilung wieder.

Der wesentliche Nachteil der Kerndichte Schätzer ist die Ausführungsgeschwindigkeit. Das entwickelte Programm benötigt zum Berechnen eines einzigen Wertes in drei Dimensionen auf der verwendeten HP Workstation bereits über eine Woche. Aus diesem Grund ist es unmöglich, an dieser Stelle Ergebnisse in der selben Ausführlichkeit wie bei den anderen Algorithmen zu präsentieren.

Gerade bei der Kerndichte Schätzung genügt aber zur Verifizierung des Algorithmus die Überprüfung der geschätzen Wahrscheinlichkeitsdichteverteilungen. Da im Phasenraum mit äquidistanter Abtastung gearbeitet wird, müssen nach dem Schätzvorgang nur noch die erhaltenen Werte nach Gleichung (3.48) verarbeitet und aufsummiert werden.

3.3.4 Vereinfachter Algorithmus

Ein gravierender Nachteil des in den vorigen beiden Kapitel entwickelten Algorithmus ist die Ausführungsgeschwindigkeit. Besonders die exponentielle Abhängigkeit von der Dimension des Phasenraumes macht das Programm für eine praktische Verwendung unbrachbar.

Um dieser Problematik zu entgehen, schlägt Matt Kennel einen modifizierten Algorithmus vor, den MMI–Algorithmus. Nach dieser Anleitung wurde ein Programm erstellt, zur Verifikation wurde parallel dazu auch die Implementierung von Kennel verwendet. In der Gleichung (2.30) wird zwei Mal die Verbundwahrscheinlichkeitsdichte verwendet. Statt beide Male den aus den Kerndichte Schätzern erhaltenen Wert zu verwenden, setzen wir vor dem Logarithmus für $p(\vec{x},\vec{y})$ die grobe Näherung

$$p(\vec{x},\vec{y}) = \frac{1}{N}\sum_{i=1}^{N}\delta(\vec{x}-\vec{x}_i)\delta(\vec{y}-\vec{y}_i) \tag{3.50}$$

ein, die jedem Datenpunkt die Wahrscheinlichkeit $1/N$ zuordnet.

Durch diese Wahl hat die geschätzte Verbundwahrscheinlichkeitsdichtefunktion nur mehr an den Datenpunkten selbst Werte ungleich Null. Daher muß nicht mehr, wie im zuvor beschriebenen Algorithmus, eine numerische Integration über den gesamten Datenraum durchgeführt werden, sondern es genügt, die zu schätzende Funktion an den Stellen auszuwerten, an denen die Datenpunkte liegen (ein Gebiet mit der Verbundwahrscheinlichkeit Null liefert keinen Beitrag zur Transinformation und muß daher auch bei der Berechnung nicht berücksichtigt werden).

Der Algorithmus wird damit vereinfacht und folgender Ablauf wird durchgeführt: es wird ein Datenpunkt aus den Eingangsdaten herausgegriffen. Für diesen Vektor werden die Beiträge aller anderen Datenpunkte zur Summenfunktion $\hat{f}(\vec{x})$ berechnet und aufsummiert. Die Summation wird dabei sowohl für die Verbundwahrscheinlichkeit als auch für die Randverteilungen durchgeführt. Man erhält so die geschätzten Wahrscheinlichkeitsdichten bei diesem Datenpunkt. Dieser Schritt wird für alle Punkte durchgeführt. Anschließend ist lediglich die Summe

$$I(X;Y) = \frac{1}{N} \sum_{i=1}^{N} \log \frac{p(\vec{x}_i, \vec{y}_i)}{p(\vec{x}_i) p(\vec{y}_i)} \tag{3.51}$$

zu bilden um die Transinformation $I(X;Y)$ zu berechnen.

Bei der Analyse der Ergebnisse wurde festgestellt, daß mit dem so vereinfachten Algorithmus falsche Ergebnisse errechnet werden. Dies kann folgendermaßen begründet werden:

- Die aus den überlagerten Kernfunktionen gebildete Wahrscheinlichkeitsdichtefunktion wird nur dort abgetastet, wo sich ein Datenpunkt befindet.
- Da die Datenpunkte nicht gleichmäßig verteilt sind, erfogt in Gebieten mit höherer Wahrscheinlichkeitsdichte auch eine dichtere Abtastung
- Der Abtastwert wird als Wahrscheinlichkeit interpretiert, es handelt sich aber um eine Wahrscheinlichkeitsdichte. Es müßte eine Skalierung proportional zur Abtastdichte erfolgen. Dies ist allerdings nicht möglich, weil die exakten Intervallgrößen unbekannt sind.

  Aus diesem Grund ist das Ergebnis des MMI–Algorithmus falsch.
- In den Bereichen mit geringer Wahrscheinlichkeitsdichte werden die Kernfunktionen nicht hinreichend oft abgetastet, um ein exaktes Ergebnis zu ermöglichen. Um genügend oft abzutasten, müssen bei jedem Datenpunkt die Beiträge der Kernfunktionen der Nachbarpunkte noch meßbar sein. Dies ist nur dann gewährleistet, wenn die Anzahl der Eingangsdaten deutlich erhöht wird. In diesem Fall ist der Algorithmus allerdings wegen der hohen Rechenzeit nicht mehr verwendbar.

Als Schlußfolgerung aus den obigen Punkten kann festgestellt werden:
Der MMI–Algorithmus von Matt Kennel ist für die Berechnung der Transinformation von Zeitreihen nicht verwendbar. Die Ergebnisse lassen keine Interpretation zu.

3.3.4.1 Die algorithmische Umsetzung

Eine Übersicht über den Ablauf der Berechnung ist im Bild 3.16 dargestellt. In den folgenden Absätzen werden die einzelnen Schritte im Detail erklärt.

Aus den skalaren Eingangsdaten wird der Mittelwert und die Energie der beiden Eingangssignale errechnet. Bei der Bildung der Vektoren aus den Eingangsdaten werden diese auf Mittelwert $\mu = 0$ und Energie $E = 1$ normiert. Durch diese Skalierung können die Kernbreiten unabhängig von den Eingangssignalen in absoluten Zahlenwerten angegeben werden. Die Vektoren werden in einer einfach verketteten Liste gespeichert.

Nun wird für jeden Eingangsvektor $\vec{x}_i$ die Liste aller Vektoren einmal durchlaufen. Bei der Bearbeitung des Vektors $\vec{x}_j$ wird der Betrag der Kernfunktion

$$\frac{1}{h^n} K\left(\frac{|\vec{x}_i - \vec{x}_j|}{h}\right) \tag{3.52}$$

errechnet. Die Dimension wird durch n bestimmt, h bezeichnet die Kernbreite. Diese Berechnung erfolgt sowohl für die beiden Randverteilungen als auch für die Verbundwahrscheinlichkeitsdichte, wobei für n die jeweilige Dimension eingesetzt wird. Nach der Summierung der einzelnen Beiträge sowie einer Skalierung erhält man für den Vektor $\vec{x}_i$ den Schätzwert

$$\hat{p}(\vec{x}_i) = \frac{1}{N}\frac{1}{h^n}\sum_{j=1}^{N} K\left(\frac{|\vec{x}_i - \vec{x}_j|}{h}\right), \tag{3.53}$$

für $\hat{p}(\vec{y}_i)$ und $\hat{p}(\vec{x}_i, \vec{y}_i)$ gelten die entsprechenden Formeln um $\vec{y}_i$ erweitert.

Nachdem alle Vektoren auf diese Weise verarbeitet worden sind, wird in einem weiteren Durchlauf der verketteten Liste für jeden Datenpunkt der Beitrag zur Transinformation berechnet. Aus der Summe der Einzelbeiträge wird das Ergebnis berechnet, nämlich

$$I(X;Y) = \sum_{i=1}^{N} \hat{p}(\vec{x}_i, \vec{y}_i) \log \frac{\hat{p}(\vec{x}_i, \vec{y}_i)}{\hat{p}(\vec{x}_i)\hat{p}(\vec{y}_i)}. \tag{3.54}$$

Um die Kernbreite nicht konstant für alle Vektoren wählen zu müssen, bietet der Algorithmus die Möglichkeit der iterativen Berechnung. Wird diese nicht verwendet, so liegt das Endergebnis bereits vor und der Algorithmus wird beendet.

Bei der iterativen Berechnung erfolgt nach dem ersten Durchlauf, in dem für alle Vektoren dieselbe Kernbreite verwendet wurde, ein zweiter Durchgang. Diesmal werden die Kernbreiten an die geschätzte Wahrscheinlichkeitsdichteverteilung angepaßt. Die Anpassung erfolgt über das geometrische Mittel g der einzelnen Schätzwerte,

$$\log g = \frac{1}{N}\sum_{i=1}^{N} \log \hat{p}(\vec{x}_i, \vec{y}_i). \tag{3.55}$$

Die adaptiven Kernbreiten h_i werden nun mit

$$h_i = h\sqrt{\frac{g}{\hat{p}(\vec{x}_i, \vec{y}_i)}} \tag{3.56}$$

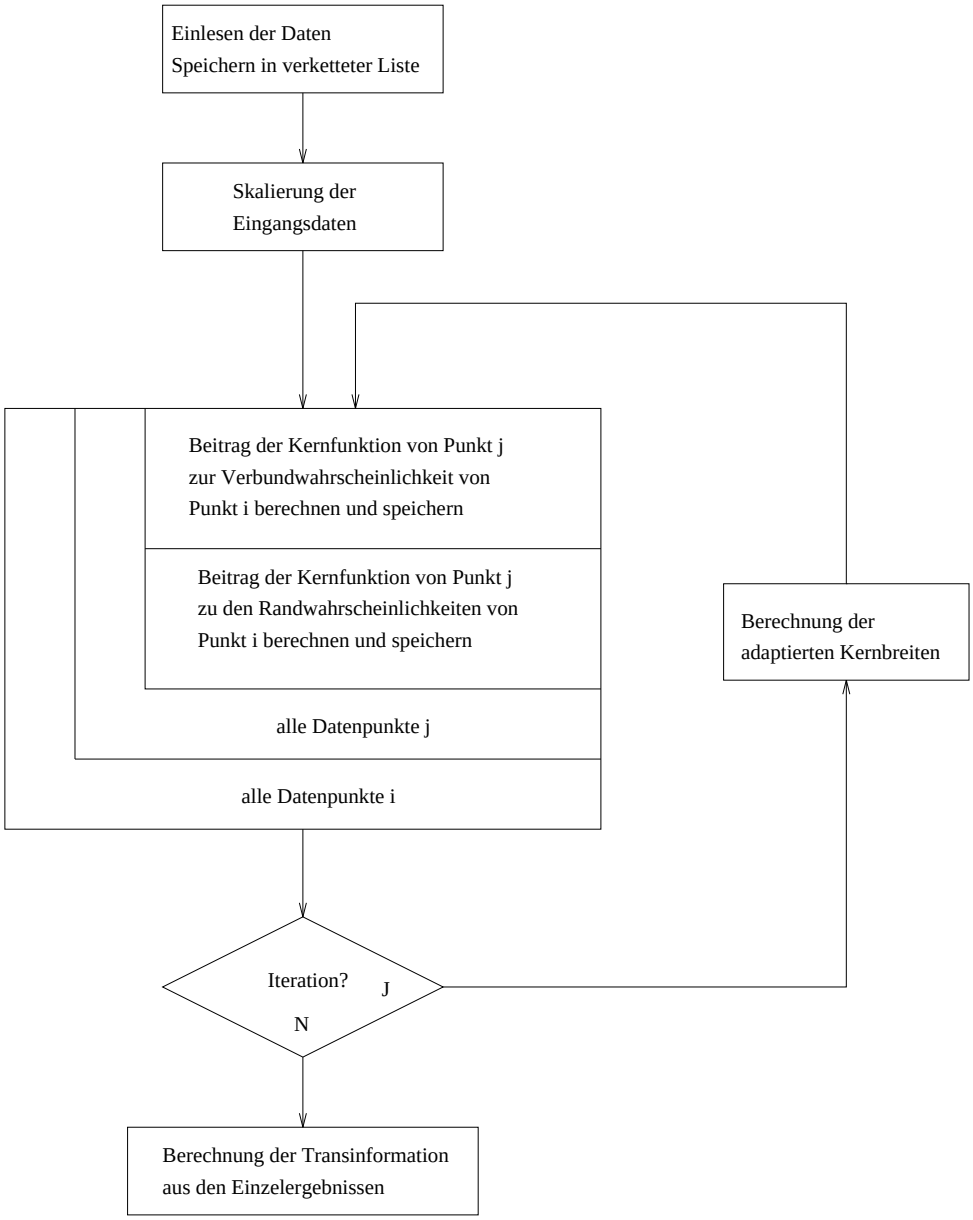


Abbildung 3.16: Ablaufdiagramm des MMI–Algorithmus

Datenpunkte	n = 1	n = 2	n = 3	n = 4	n = 5
1k	3,97	4,34	4,55	4,69	4,82
2k	3,96	4,33	4,52	4,67	4,79
4k	3,96	4,33	4,53	4,67	4,79
8k	3,96	4,33	4,53	4,67	4,79
16k	3,96	4,33	4,53	4,67	4,79
32k	3,96	4,33	4,53	4,67	4,78

Tabelle 3.5: Transinformation in Bit einer mit SNR = 42 dB verrauschten Sinusschwingung, berechnet mit dem MMI–Algorithmus

berechnet, wobei h der in der ersten Iteration verwendete Wert ist. Mit diesen Kernbreiten wird nun der gesamte Vorgang wiederholt, die neuen Schätzwerte der Wahrscheinlichkeitsdichte $\hat{p}(\vec{x}_i)$ berechnen sich aus

$$\hat{p}(\vec{x}_i) = \frac{1}{N} \sum_{j=1}^{N} \frac{1}{{h_i}^n} K\left(\frac{|\vec{x}_i - \vec{x}_j|}{h}\right). \tag{3.57}$$

Für $\hat{p}(\vec{y}_i)$ und $\hat{p}(\vec{x}_i, \vec{y}_i)$ gilt (3.57) sinngemäß.

3.3.4.2 Ergebnisse

Die in Tabelle 3.5 zusammengefaßten Werte für die Transinformation sind mit dem bereits früher verwendeten, mit 42 dB verrauschten Sinussignal erhalten worden. Bei der Berechnung wurde die Anzahl der Eingangsdatenpunkte und die bei der Konstruktion des Phasenraumes verwendete Dimension variiert. Es fällt auf, daß die Ergebnisse deutlich vom geforderten Wert von 7 Bit abweichen und damit die weiter oben gebrachten Argumente, daß der MMI–Algorithmus falsche Werte errechnet, bestätigen.

Zusätzlich zur groben Abweichung fällt auf, daß die berechnete Transinformation praktisch unabhängig von der Anzahl der verwendeten Datenpunkte ist. Diese Eigenschaft ist an sich wünschenswert, allerdings in diesem Ausmaß bei funktionierenden Algorithmen nicht erzielbar, da es bei starker Verringerung der Menge der Eingangsdaten grundsätzlich nicht möglich ist, eine gute Schätzung der Wahrscheinlichkeitsdichteverteilung zu erzielen, unabhängig vom verwendeten Algorithmus.

Kapitel 4

Ergebnisse

4.1 Gauß'sches Rauschen

4.1.1 Definition

Die Verteilungsdichtefunktion von weißem gaußschen Rauschen ist durch

$$p_N(x) = \frac{1}{\sqrt{2\pi}\,\sigma} e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}} \tag{4.1}$$

gegeben. Zur Überprüfung der entwickelten Programme werden zwei Zeitreihen $n_1(t)$ und $n_2(t)$ verwendet. Bei beiden handelt es sich um weißes gaußsches Rauschen mit einem Mittelwert $\mu = 0$ und einer Varianz $\sigma = 1$, die Wahrscheinlichkeitsdichteverteilungen sind N_1 und N_2. Mit dem Ergebnis kann eine erste Aussage über die Korrektheit des implementierten Verfahrens getroffen werden. Die Transinformation

$$I(N_1; N_2) = 0 \tag{4.2}$$

muß verschwinden, da in den bereits bekannten Werten des verrauschten Signales keine Information über zukünftige Werte enthalten ist. Für die Kullback Leibler Distanz gilt ebenfalls

$$D(N_1||N_2) = 0, \tag{4.3}$$

da die Verteilungsdichtefunktionen der beiden Signale identisch sind.

Für die Schätzung der Gaußverteilung sind verhältnismäßig viele Datenpunkte nötig, um ein genaues Ergebnis zu erhalten. Das ist der Fall, da bei Dimensionen größer als eins keine Trajektorien vorhanden sind, auf denen die Datenpunkte liegen. Statt dessen sind die Vektoren auf ein relativ großes Volumen verteilt.

4.1.2 Referenzergebnisse

Um eine Bewertung der Ergebnisse durchführen zu können, werden die Resultate der neu entwickelten Programme mit denen des FMI–Algorithmus von H.-P. Bernhard [1] verglichen. Für gaußsches Rauschen sind die Ergebnisse des FMI–Algorithmus in Abbildung 4.1 dargestellt. Es wurde die Transinformation für unterschiedliche Mengen

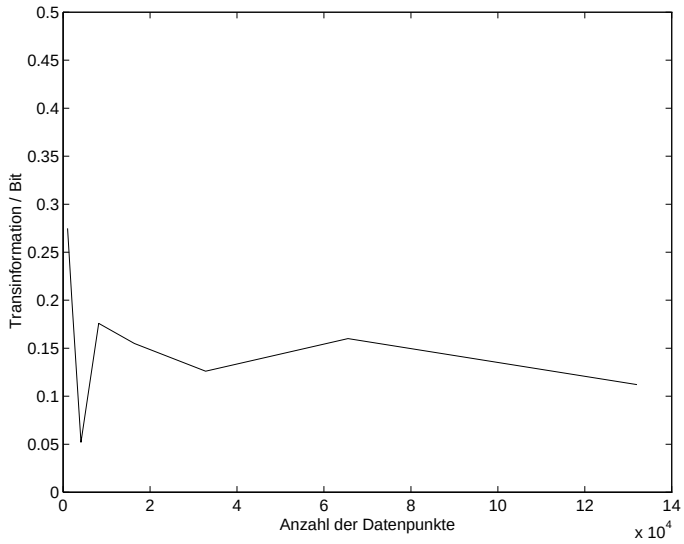


Abbildung 4.1: Transinformation von gaußschem Rauschen, berechnet mit dem FMI–Algorithmus

an Datenpunkten ermittelt. Man erkennt eine leichte Abnahme des berechneten Wertes bei einer größeren Anzahl an Datenpunkten. Der Zahlenwert des Ergebnisses liegt etwa bei $0,15$ Bit. Die Abweichung vom theoretischen Ergebnis $I(N_1; N_2) = 0$ entsteht durch Schätzfehler. Die Größe der Abweichung kann als Maßstab für die Rechengenauigkeit des Algorithmus verwendet werden. Die Ergebnisse der neu entwickelten Verfahren werden daher mit dem Ergebnis des FMI–Algorithmus verglichen.

4.1.3 Adaptives Histogramm

Gaußverteilte Wahrscheinlichkeitsdichtefunktionen sind für die adaptive Histogrammethode ein besonders geeigneter Test. Bei der Berechnung erfolgt, wie in Kapitel 3.2.2 im Detail erläutert wird, nach jeder Iteration eine Überprüfung, ob lokal Gleichverteilung vorliegt. Dies entspricht einem flachen Bereich in der Wahrscheinlichkeitsdichteverteilung. Da in der Gaußverteilung solche Bereiche nicht enthalten sind, benötigt der Algorithmus besonders viele Rekursionen, um eine geeignete Schätzung zu finden. Im Gegensatz dazu würde bei gleichverteiltem Rauschen bereits bei der zweiten Rekursion abgebrochen werden.[1]

4.1.3.1 Transinformation

In Bild 4.2 sind die Ergebnisse der Berechnungen für die Transinformation zusammengestellt. Es ist zu erkennen, daß die geschätzte Transinformation abnimmt, je mehr

[1] Die erste Rekursion wird auf jeden Fall durchgeführt, unabhängig von der Verteilungsdichtefunktion, wie in Kapitel 3.2.1.3 beschrieben.

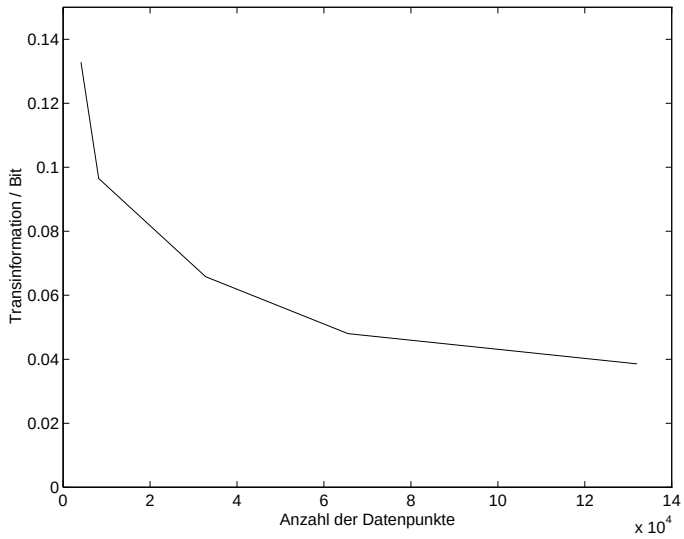


Abbildung 4.2: Transinformation von gaußschem Rauschen, berechnet mit adaptiven Histogrammen

Datenpunkte für die Berechnung verwendet werden. Dies liegt an der besseren Rekonstruktion der Verbundwahrscheinlichkeitsdichte, je mehr Datenpunkte zur Verfügung stehen.

Der Zahlenwert für die errechnete Transinformation liegt etwa bei $0,06$, das ist eine deutliche Verbesserung gegenüber den vorher gezeigten Werten, die mit dem FMI–Algorithmus erhalten werden.

4.1.3.2 Kullback Leibler Distanz

Ähnlich wie bei der Transinformation zeigt sich, wenn man die Kullback Leibler Distanz von weißem gaußschem Rauschen berechnet, daß das Ergebnis umso genauer wird, je mehr Eingangsdaten für die Berechnung verwendet werden. Die Ergebnisse dieser Berechnung sind in Abbildung 4.3 zusammengefaßt.

Die Kullback Leibler Distanz ist Null, falls die Wahrscheinlichkeitsdichtefunktionen der beiden Eingangssignale identisch sind. Das bedeutet, daß nicht nur bei Rauschsignalen, sondern bei beliebigen Signalen mit identischer Wahrscheinlichkeitsdichteverteilung das Ergebnis Null sein sollte.

4.1.4 Kerndichte Schätzung

Beim gaußschen Rauschen werden außer der zweidimensionalen Verbundwahrscheinlichkeitsdichte (Abbildung 4.5) auch die Schätzwerte für die Randverteilungen dargestellt (Abbildung 4.4). Man kann deutlich die Gaußverteilung erkennen, es sind aber merkbare Unregelmäßigkeiten vorhanden. Diese entstehen, weil nur 7000 Datenpunkte berücksichtigt werden. Für die Gauß–Verteilung liegt diese Zahl an der unteren Tole-

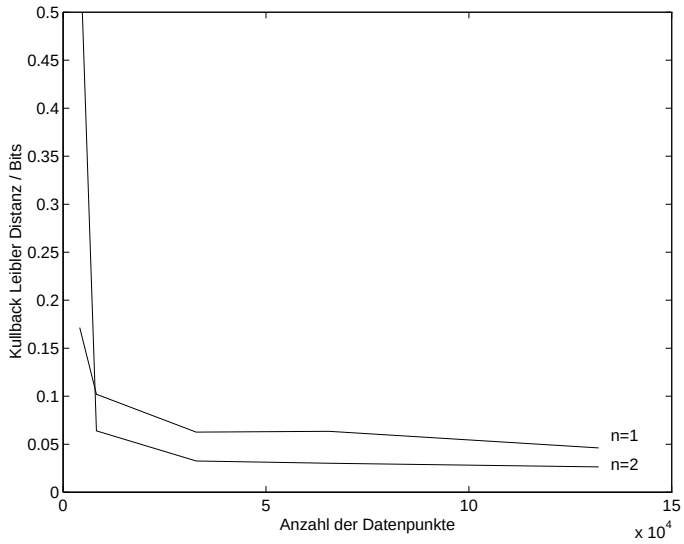


Abbildung 4.3: Kullback Leibler Distanz von gaußschem Rauschen, berechnet mit adaptiven Histogrammen

ranzgrenze (im Gegensatz zu den anderen Systemen, wie im Anschluß gezeigt wird).

Die Ergebnisse des MMI–Algorithmus bei der Berechnung der Transinformation von weißen Rauschen sind in Abbildung 4.6 zusammengefaßt. Man erkennt sofort, daß die Zahlenwerte wesentlich vom erwarteten Ergebnis abweichen, das Ergebnis ist etwa um einen Faktor 30 größer als beim Referenzalgorithmus.

4.2 Quasiperiodisches Signal

4.2.1 Definition

Der zweite Test erfolgt mit einem verrauschten, quasiperiodischen Signal. Bereits die Algorithmen aus [7] und [1] wurden mit einer Überlagerung von vier Sinusfunktionen getestet. Da für dieses Signal auch die Referenzergebnisse vorliegen, wird es zum Testen der neu entwickelten Algorithmen eingesetzt.

Als quasiperiodisches Testsignal dient

$$x(t) = sin(\omega_1 t) + \frac{sin(3\omega_1 t)}{9} + \frac{sin(\omega_2 t)}{4} + \frac{sin(3\omega_2 t)}{36} + N(t). \tag{4.4}$$

Dabei ist $N(t)$ weißes Rauschen mit einem Signal/Rauschverhältnis von etwa 42 dB, $\omega_1 = 6/40$, $\omega_2 = \omega_1(1+\sqrt{5})/10$ und t ist ein ganzzahliger Zeitparameter. Die Verzögerungszeit T bei der Konstruktion der Vektoren wird $T = 20$ gewählt.

Als Ergebnis sollte unabhängig von der zeitlichen Verschiebung der Eingangssignale ein konstanter Wert erhalten werden, da eine Voraussage über Werte in der nahen Zukunft bei diesem Signal mit derselben Sicherheit gemacht werden kann, wie eine

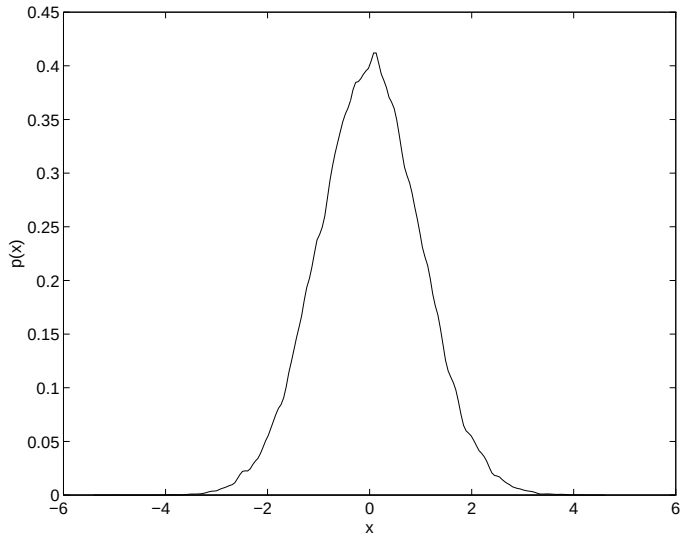


Abbildung 4.4: Kerndichteschätzung von gaußschem Rauschen in einer Dimension

Originaldaten *Kerndichte Schätzung*

Abbildung 4.5: Kerndichteschätzung von gaußschem Rauschen in zwei Dimensionen

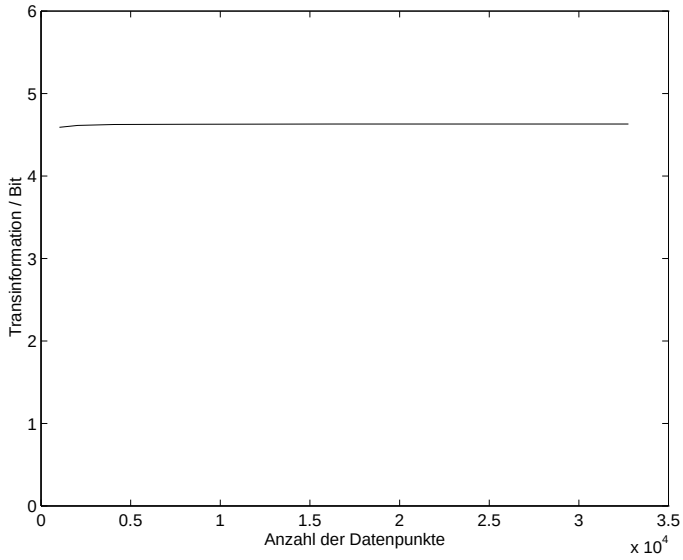


Abbildung 4.6: Transinformation von gaußschem Rauschen, berechnet mit dem MMI–Algorithmus

Voraussage über beliebig entfernte Zeitpunkte. Die Genauigkeit der Prädiktion wird nur durch das zugefügte Rauschen limitiert, das Ergebnis für die Transinformation ist daher 42 dB, das entspricht einer Transinformation

$$I(X;Y) = \frac{42\,dB}{6,02\,dB/Bit} \approx 7\,Bit. \tag{4.5}$$

Durch diese Berechnung wird auch klar, warum das additive Rauschen notwendig ist. Bei einem unverrauschten Signal würde die Transinformation über alle Grenzen steigen, da die exakte Vorhersage des Signalverlaufes möglich ist.[2]

4.2.2 Referenzergebnisse

Die mit dem FMI–Algorithmus für das quasiperiodische Signal (4.4) ermittelten Werte werden in Abbildung 4.7 gezeigt, es wird sowohl die Anzahl der Datenpunkte als auch die Dimension variiert. Man erkennt, daß sich das Ergebnis dem Sättigungswert nähert, wenn mehr Eingangsdaten zur Verfügung stehen. Wird die Dimension erhöht, so tendiert der berechnete Wert ebenfalls gegen 7 Bit. Allerdings kann es bei hohen Dimensionen ($n \geq 4$) aus dem folgenden Grund zu Schwankungen in der Schätzung kommen: Die Daten konzentrieren sich mit steigender Dimension auf immer kleinere Bereiche ([12], dort wird diese Erscheinung als 'Fluch der Dimensionalität' bezeichnet). Es sind immer mehr Datenpunkte nötig, damit der Algorithmus die Struktur der Ver-

[2]Im Gegensatz zu chaotischen Signalen, wo auch bei unverrauschten Systemen keine exakte Prädiktion erzielt werden kann

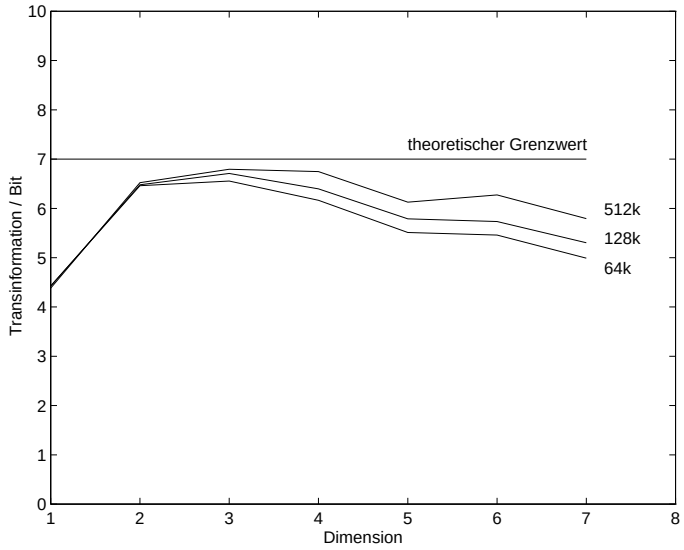


Abbildung 4.7: Transinformation des quasiperiodischen Signals (4.4), berechnet mit dem FMI–Algorithmus

teilungsdichtefunktion korrekt erkennen kann.[3] Je nach Lage der Anhäufungen kann es vorkommen, daß die Transinformation über- oder unterschätzt wird.

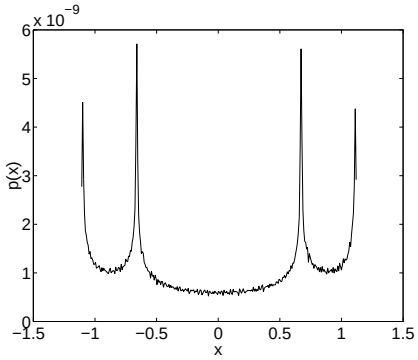
4.2.3 Kerndichte Schätzung

Die mittels Kerndichte Schätzung berechneten Wahrscheinlichkeitsdichteverteilungen für eine bzw. zwei Dimensionen sind in den Bildern 4.8 und 4.9 dargestellt. Links befinden sich jeweils mit den Originaldaten erzeugten Diagramme, das rechte Teilbild zeigt die mit Kerndichte Schätzern ermittelten Ergebnisse.

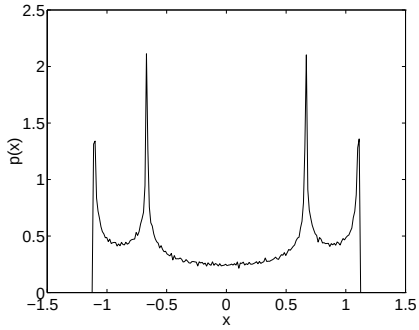
In Abbildung 4.10 sind die Ergebnisse des MMI–Algorithmus für das quasiperiodische Eingangssignal zusammengestellt. Man erkennt, daß das Ergebnis deutlich vom mathematisch exakten Wert abweicht. Es ist weiters zu bemerken, daß bis zu Dimensionen $n = 5$ der berechnete Wert weder von der Dimension noch von der Anzahl der Eingangsdaten stark beeinflußt wird.

Bei $n > 5$ treten starke Schwankungen im Ergebnis auf, es kommt sogar teilweise zu negativen Resultaten. Diese Probleme hängen allerdings mit der zu geringen Anzahl an Eingangsdaten zusammen und sind nicht ausschließlich auf den Algorithmus zurückzuführen. Eine Berechnung mit mehr Datenpunkten konnte wegen der hohen Rechenzeiten nicht durchgeführt werden.

[3] Um in vier Dimensionen die selbe Datendichte zu erreichen, wie bei einer eindimensionalen Schätzung mit 100 Datenpunkten, sind bereits 10^6 Werte notwendig.

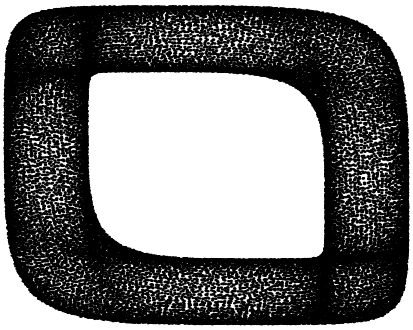


Originaldaten

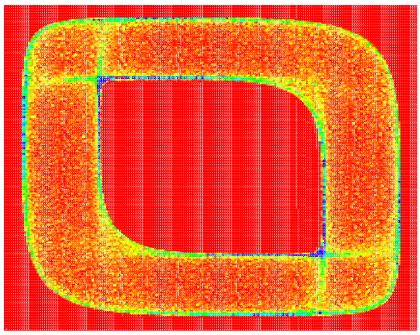


Kerndichte Schätzung

Abbildung 4.8: Quasiperiodisches Signal (4.4) in einer Dimension



Originaldaten



Kerndichte Schätzung

Abbildung 4.9: Quasiperiodisches Signal (4.4) in zwei Dimensionen

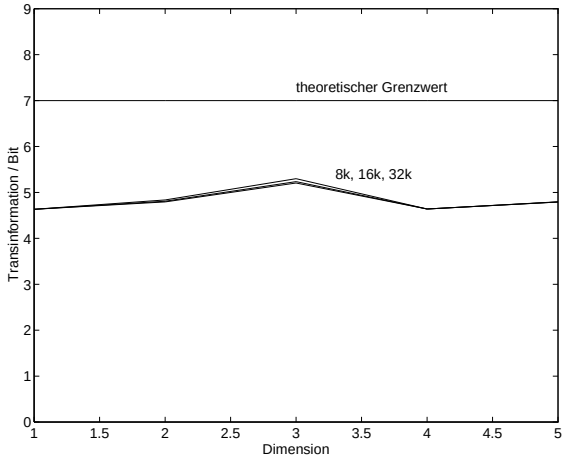


Abbildung 4.10: Transinformation des quasiperiodischen Signals (4.4), berechnet mit dem MMI–Algorithmus

4.3 Rösslersystem

4.3.1 Definition

Das Rösslersystem ist ein chaotisches System, das durch die Differentialgleichungen

$$\begin{aligned} \dot{x} &= -z - y \\ \dot{y} &= x + ay \\ \dot{z} &= b + z(x - c) \end{aligned} \tag{4.6}$$

beschrieben wird (siehe [4]). Die Parameter werden

$$\begin{aligned} a &= 0,15 \\ b &= 0,20 \\ c &= 10,0 \end{aligned} \tag{4.7}$$

gewählt. Die Zeitverzögerung T zur Konstruktion der Vektoren nach (2.1) beträgt 8.

Die Trajektorie des Rösslersystems weist eine Struktur auf, die mit der euklidischen Geometrie nicht beschreibbar ist. Man findet, je genauer man die Wahrscheinlichkeitsdichteverteilung betrachtet, immer feinere Detailstrukturen. Das Kriterium für einen guten Algorithmus zur Berechnung der Transinformation ist die Fähigkeit, mit diesen starken Schwankungen umgehen zu können.

Im Rösslersystem wird, da es chaotisches Verhalten zeigt, laufend Information erzeugt. Man kann wie bei jedem System, wenn die Funktionswerte der Vergangenheit bekannt sind, Prädiktionen über den zukünftigen Funktionsverlauf machen. Die Sicherheit dieser Schätzungen nimmt aber ab, je weiter der zu schätzende Zeitpunkt in der Zukunft liegt.

Auf die Transinformation angewandt bedeutet das:

- Die Transinformation wird für geringe Zeitverzögerungen hoch sein (Prädiktion der nahen Zukunft) und mit zunehmender Verzögerung sinken.
- Mit steigender Dimension (entspricht der Anzahl an bekannten Funktionswerten aus der Vergangenheit) nimmt auch die Transinformation zu (bessere Prädiktion ist möglich). Allerdings wird ein Sättigungswert erreicht, sobald die in der Berechnung verwendete Dimension die fraktale Dimension des Rösslersystems übersteigt.

4.3.2 Referenzergebnisse

Die Ergebnisse des Referenzalgorithmus sind in Abbildung 4.11 dargestellt. Die Berechnung erfolgt für $2 \rightarrow 1$ und für $3 \rightarrow 1$ Dimension. Die Eingangsdaten für die Zufallsvariable Y werden gegenüber denen für die Zufallsvariable X von $0 - 200$ Zeitschritte verzögert. Man in der Grafik deutlich die Abnahme der Transinformation mit zunehmender Verzögerung erkennen. Die Steigung der Kurve ist ein Maß für die Informationsproduktion des Rösslersystems und somit als Beurteilungskriterium für die entwickelten Programme verwendbar.

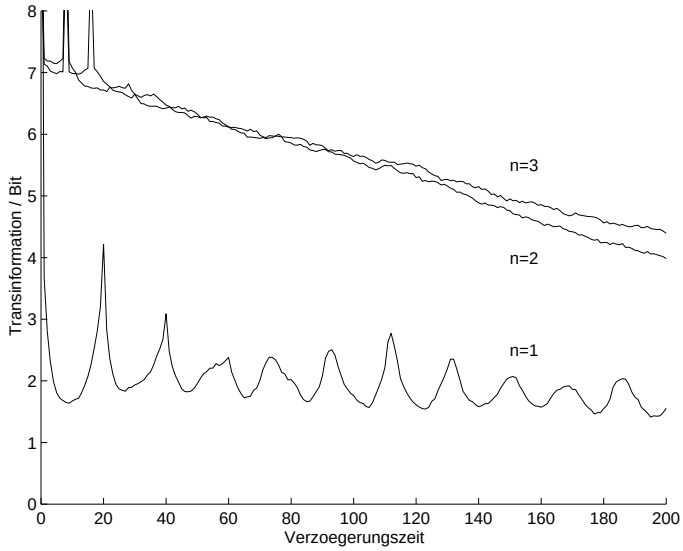


Abbildung 4.11: Transinformation eines Rösslersystems, berechnet mit dem FMI–Algorithmus

4.3.3 Adaptives Histogramm

In Abbildung 4.12 sind Ergebnisse der durchgeführten Berechnungen dargestellt. Ein Vergleich mit Abbildung 4.11 zeigt, daß die Struktur der Ergebnisse gleich ist.

Auch die Steigung der Kurve, aus der man die Rate der Informationsproduktion schätzen kann, ist identisch mit der des Referenzprogrammes.

4.3.4 Kerndichte Schätzung

Das linke Teilbild von Abbildung 4.13 zeigt ein Streudiagramm der verwendeten Eingangsdaten, im rechten Teilbild wird die mit Kerndichte Schätzung ermittelte Wahrscheinlichkeitsdichteverteilung dargestellt. Man kann deutlich die Übereinstimmung zwischen Gebieten, die viele Datenpunkte enthalten, beim Streudiagramm und Gebieten mit hoher Wahrscheinlichkeitsdichte bei der Kerndichte Schätzung erkennen.

Die mit dem MMI–Algorithmus errechneten Werte weichen auch beim Rössler–System erheblich von den vorgegebenen Ergebnissen ab (siehe Bild 4.14). Für $n = 1$ stimmen die Resultate zwar mit denen des Referenzprogrammes überein, für $n = 2$ und $n = 3$ zeigt sich jedoch ein völlig falsches Verhalten. Sowohl der Zahlenwert für die Transinformation, als auch die Steigung der Kurve sind verschieden von den Ergebnissen der beiden anderen Algorithmen.

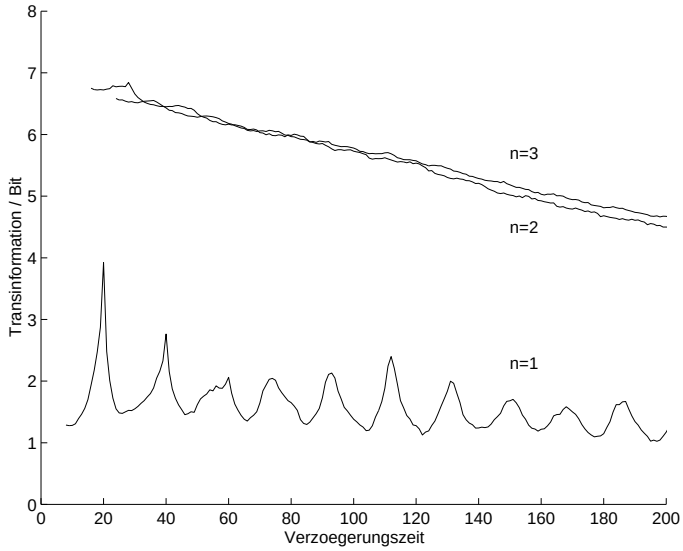


Abbildung 4.12: Transinformation eines Rösslersystems, berechnet mit adaptiven Histogrammen

Originaldaten *Kerndichte Schätzung*

Abbildung 4.13: Kerndichteschätzung des Rösslersystems in zwei Dimensionen

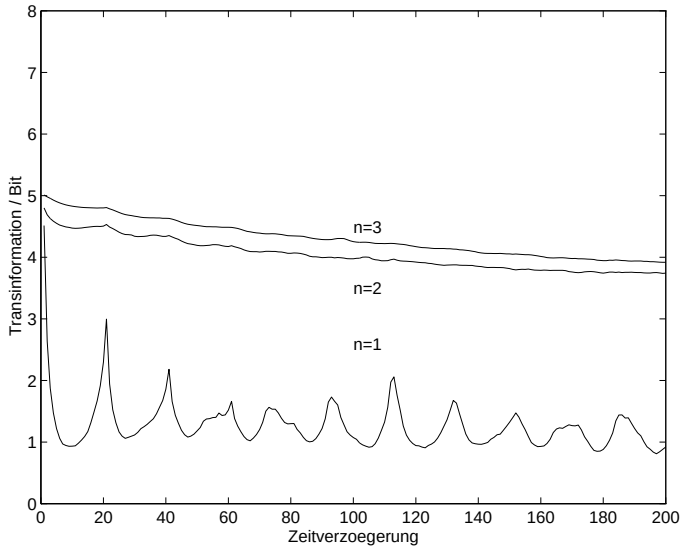


Abbildung 4.14: Transinformation eines Rösslersystems, berechnet mit dem MMI–Algorithmus

4.4 Lorenzsystem

4.4.1 Definition

Beim Lorenzsystem handelt es sich wie beim Rösslersystem um ein chaotisches System. Es wird durch die Differentialgleichungen

$$\begin{aligned} \dot{x} &= \sigma(y - x) \\ \dot{y} &= x(R - z) - y \\ \dot{z} &= b + z(x - c) \end{aligned} \tag{4.8}$$

beschrieben. Die Parameter sind

$$\begin{aligned} \sigma &= 16,0 \\ R &= 45,92 \\ b &= 4,0. \end{aligned} \tag{4.9}$$

Die Zeitverzögerung T zur Konstruktion der Vektoren nach (2.1) beträgt 15.

4.4.2 Referenzergebnisse

Die Ergebnisse des FMI–Algorithmus sind in Abbildung 4.15 dargestellt.

4.4.3 Adaptives Histogramm

Die Ergebnisse der Berechnung des Lorenzsystems mit adaptiven Histogrammen sind in Bild 4.16 zusammengefaßt. Es gelten die selben Bemerkungen wie beim Rösslersystem.

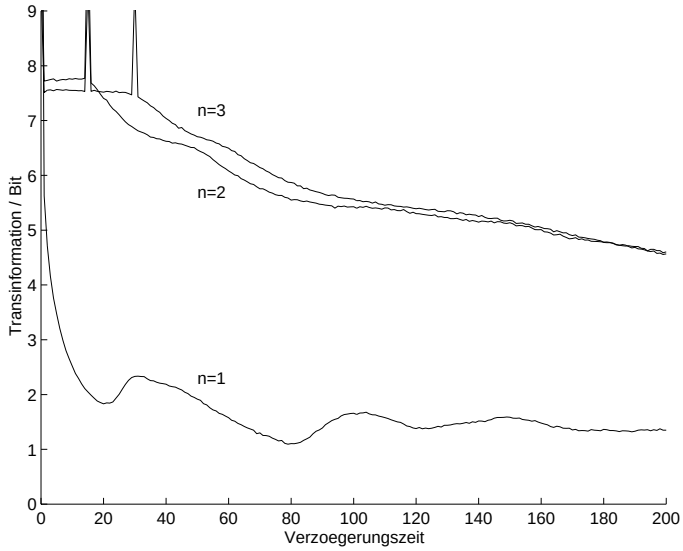


Abbildung 4.15: Transinformation eines Lorenzsystems, berechnet mit dem FMI–Algorithmus

Die Kurven sind qualitativ identisch mit dem Referenzsystem.

4.4.4 Kerndichte Schätzung

In Abbildung 4.13 wird, ähnlich wie beim Rösslersystem, ein Streudiagramm der Eingangsdaten (linkes Teilbild) der mit Kerndichte Schätzung ermittelten Wahrscheinlichkeitsdichteverteilung (rechtes Teilbild) gegenübergestellt. Auch hier ist die gute Übereinstimmung zwischen den beiden Darstellungen zu bemerken.

Die Berechnung des Lorenzsystems mit dem MMI–Algorithmus wurde nicht durchgeführt, da mit den bisherigen Untersuchungen bereits hinreichend gezeigt wurde, daß keine aussagekräftigen Ergebnisse erhalten werden können.

4.5 Ausführungszeit

Ein wichtiges Kriterium für die praktische Verwendbarkeit von Algorithmen ist (abgesehen von der Genauigkeit der berechneten Ergebnisse) die Dauer der Berechnung. Meist sind dabei nicht die absoluten Rechenzeiten von Bedeutung, sondern die Abhängigkeit von wesentlichen Parametern der Aufgabenstellung. In dieser Arbeit ist wichtig, wie sich die Rechenzeit verhält, wenn die Dimension oder die Anzahl der berücksichtigten Datenpunkte verändert wird.

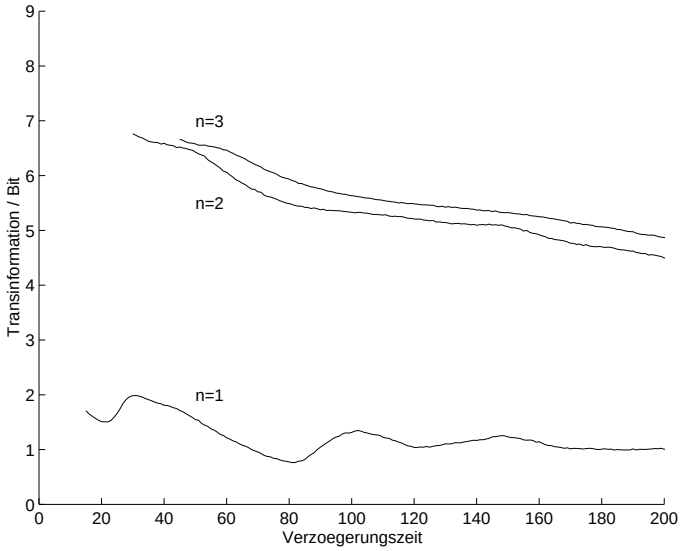


Abbildung 4.16: Transinformation eines Lorenzsystems, berechnet mit adaptiven Histogrammen

Originaldaten *Kerndichte Schätzung*

Abbildung 4.17: Kerndichteschätzung des Lorenzsystems in zwei Dimensionen

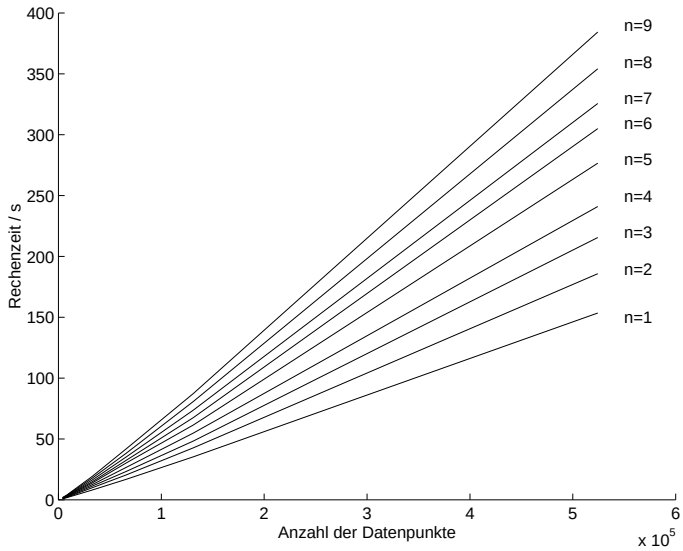


Abbildung 4.18: Rechendauer beim FMI–Algorithmus in Abhängigkeit von der Anzahl der Eingangsdaten für eine Sinusschwingung

4.5.1 Referenzwerte

Als Vergleichswerte werden die Ergebnisse angeführt, die mit dem FMI–Algorithmus von H.-P. Bernhard [1] erhalten werden. Die Ausführungszeit wird in Abhängigkeit von der Anzahl der Eingangsdaten (Bild 4.18) sowie der Dimension dargestellt (Bild4.19).

Man erkennt, daß sowohl die Dimension als auch die Anzahl der Datenpunkte etwa linear in die Rechenzeit eingehen. Da der Referenzalgorithmus eine andere Berechnungsmethode verwendet, kann die Schätzung der Randwahrscheinlichkeitsdichten entfallen. Aus diesem Grund können die hier gezeigten Ergebnisse von den neu entwickelten Programmen nicht erreicht werden.

4.5.2 Adaptives Histogramm

Die Tatsache, daß ein Histogramm datenabhängig ist (es erfolgt eine adaptive Anpassung an die Eingangsdaten), macht es schwierig, exakte Aussagen über die Rechenzeit zu treffen, da diese stark von den Eingangsdaten abhängt. Die Laufzeit des Algorithmus nach der adaptiven Histogrammethode wird durch die folgenden Parameter wesentlich beeinflußt:

- Die Anzahl der Datenpunkte. Der Algorithmus erzeugt drei Baumstrukturen (einen für die Verbundwahrscheinlichkeitsdichte und zwei für die beiden Randverteilungen). Die Eingangsdaten werden analysiert, um die Verzweigungen der Bäume festzulegen. Die Datenpunkte innerhalb eines Hyperkubus werden auf Gleichverteilung untersucht (siehe Kapitel 3.19). Liegt diese nicht vor, so wird der Hyperkubus in 2^n Subkuben unterteilt. Dazu kann folgendes festgestellt werden:

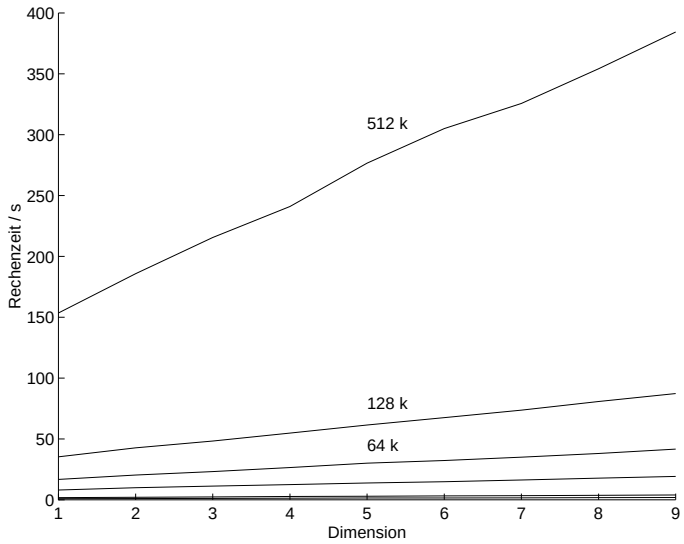


Abbildung 4.19: Rechendauer beim FMI–Algorithmus in Abhängigkeit von der Dimension für eine Sinusschwingung

Die geschätzte Wahrscheinlichkeitsdichteverteilung ändert sich nicht in Abhängigkeit von der Anzahl der Datenpunkte. Lediglich bei zu wenigen Eingangsdaten erscheint die Verteilung weniger glatt. In diesem Fall ist das Rechenergebnis allerdings ohnehin unpräzise, da die Wahrscheinlichkeitsdichte bei ungenügender Abtastung mit keinem Kriterium mehr bestimmt werden kann.

Sobald die Anzahl der Vektoren ausreichend ist, um eine Schätzung der Wahrscheinlichkeitsdichteverteilung zu ermöglichen (diese hängt unter anderem auch von der Struktur der Daten ab, kann also nicht generell angegeben werden), bleibt die Geometrie der Bäume unverändert. Nachdem alle Datenpunkte jeweils bis zu einem Blatt des Baumes klassifiziert werden (als *Blatt* wird ein Hyperkubus bezeichnet, in dem Gleichverteilung herrscht), sind bei N Datenpunkten und einer mittleren Tiefe l des Baumes Nl Operationen notwendig. Da das Zuordnen zu den entsprechenden Hyperkuben (bezüglich der Rechenzeit) im Allgemeinen die dominante Operation innerhalb des Algorithmus ist (siehe auch den nächsten Punkt), kann mit linearem Zuwachs der Ausführungszeit in Abhängigkeit von der Anzahl der Eingangsdatenpunkte gerechnet werden. Dies wird in Abbildung 4.20 bestätigt.

- Die Anzahl der Dimensionen. Wird in einem Hyperkubus eine Unterstruktur in der zu schätzenden Wahrscheinlichkeitsdichteverteilung gefunden, so werden bei n Dimensionen 2^n Subwürfel erzeugt. Die Anzahl der Daten, die in die Subwürfel verschoben werden, ist konstant, so daß an dieser Stelle kein merkbarer Einfluß auf die Rechenzeit entsteht.

  Eine exponentielle Abhängigkeit von der Anzahl der Dimensionen zeigt sich aber beim χ^2–Test, bei dem über alle 2^n Unterbereiche summiert werden muß. Au-

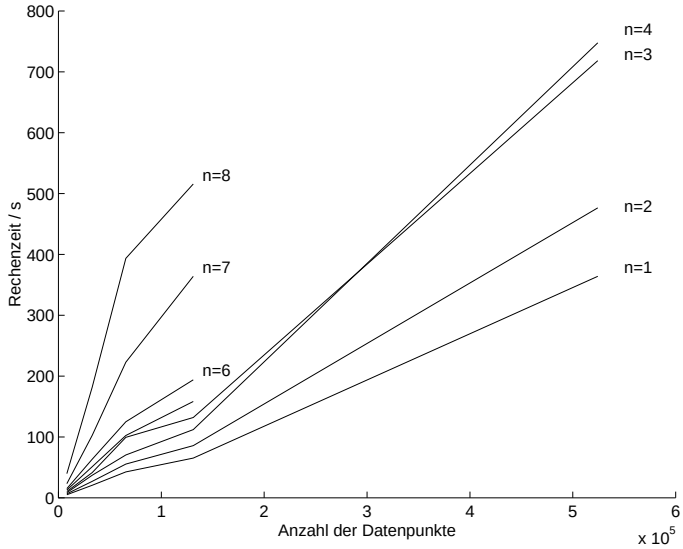


Abbildung 4.20: Rechendauer beim adaptiven Histogramm in Abhängigkeit von der Anzahl der Eingangsdaten für eine Sinusschwingung

ßerdem erfolgt, falls eine Unterstruktur vermutet wird, eine Rekursion, die sich ebenfalls über alle 2^n Bereiche erstreckt[4].

Wenn die Rekursion weit fortgeschritten ist, der Algorithmus sich also den Blättern des aufgebauten Baumes nähert, so sind in den einzelnen Kuben nur noch wenige Datenpunkte enthalten, während andererseits eine exponentiell steigende Anzahl an Kuben abzuarbeiten ist. Obwohl an und für sich das Verschieben der Daten (das von der Dimension ja unabhängig ist) mehr Rechenzeit benötigt, dominiert insgesamt daher das exponentielle Wachstum (siehe Bild 4.21).

Dabei muß auch noch berücksichtigt werden, daß sich bei höheren Dimensionen die Datenpunkte auf immer kleinere räumliche Bereiche konzentrieren, während andererseits große Teile des Raumes weitgehend leer sind. Dieses Verhalten führt zu einer Zunahme der Laufzeit, wie sie im nächsten Absatz beschrieben wird.

- Die Wahrscheinlichkeitsdichteverteilung. Der Algorithmus teilt einen Hyperwürfel nur dann in Unterklassen auf, wenn eine signifikante Struktur in der Wahrscheinlichkeitsdichteverteilung erkannt wird. Liegt also eine besonders flache Verteilung vor, so sind wesentlich weniger Unterteilungen notwendig, als wenn die zu schätzende Funktion zahlreiche Detailstrukturen aufweist. Da es kein Maß für die Dichte an Strukturen innerhalb einer Funktion gibt, können keine quantitativen Aussagen über die Abhängigkeit gemacht werden. Es wird jedoch darauf hingewiesen, daß mit verrauschten Eingangsdaten bessere Ergebnisse erzielt werden, da in diesem Fall die zu schätzende Wahrscheinlichkeitsdichteverteilung glatter

[4]Wenn in einem Unterbereich allerdings gar keine Daten mehr enthalten sind, so wird er bei der Rekursion nicht mehr berücksichtigt.

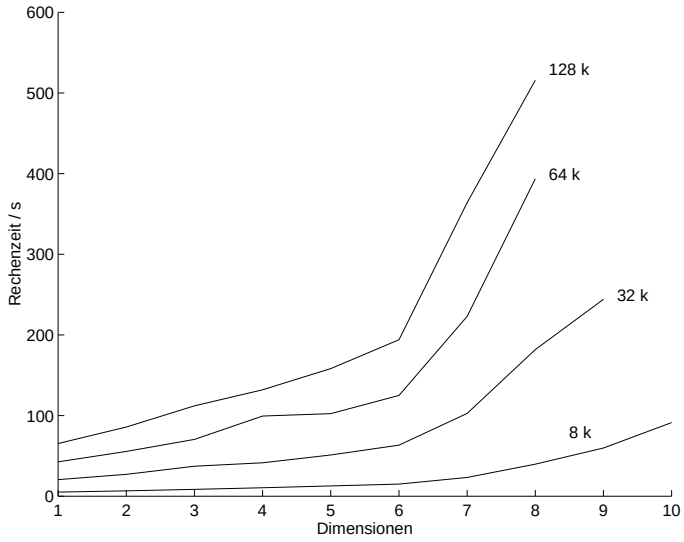


Abbildung 4.21: Rechendauer beim adaptiven Histogramm in Abhängigkeit von der Dimension für eine Sinusschwingung

ist.

4.5.3 Kerndichte Schätzung

Der Algorithmus berechnet und speichert zunächst die geschätzte Wahrscheinlichkeitsverteilung der beiden Randverteilungen, anschließend werden die einzelnen Gitterpunkte der Summenverteilung ermittelt.

- In Abhängigkeit von der Anzahl an Eingangsdaten wächst die Rechenzeit linear an, da bei jedem Gitterpunkt über alle Eingangsdaten summiert wird. Dieser Zusammenhang wird in Abbildung 4.22 dargestellt.

- Die Dimension beeinflußt die Rechenzeit exponentiell, wie in Abbildung 4.23 zu sehen ist. Wenn die beiden Randverteilungen n_1 und n_2 -dimensional sind, die Summenverteilung daher $n = n_1 + n_2$ Dimensionen besitzt und N Datenpunkte zur Verfügung stehen, so werden insgesamt (bei K Abtastpunkten pro Koordinate)

$$NK^{n_1} + NK^{n_2} + NK^n = N(K^{n_1} + K^{n_2} + K^n) \tag{4.10}$$

  Funktionswerte überlagert. Die tatsächliche Ausführungszeit hängt von der Implementation der Kernfunktion ab. Beispielsweise muß ein Kern mit endlicher Ausdehnung für entfernte Punkte nicht dieselbe Rechenzeit benötigen wie für nahegelegene.[5]

[5]Eine Vereinfachung für entfernte Punkte ergibt sich meist daraus, daß langsame arithmetische Operationen entfallen können, da ab einer gewissen Distanz das Ergebnis ohnehin identisch Null ist.

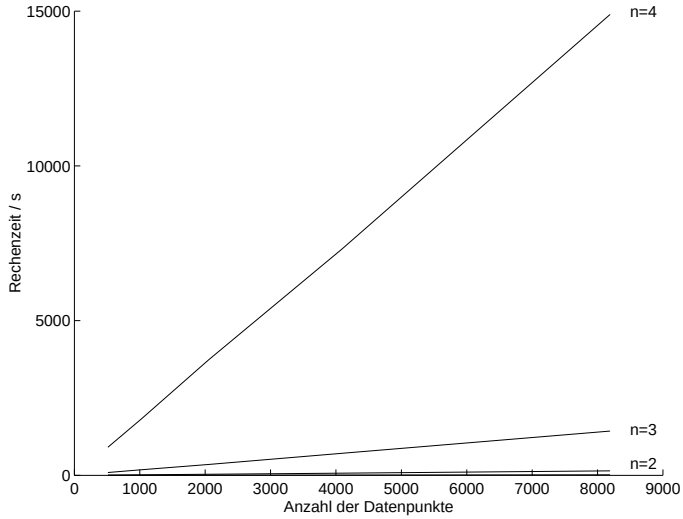


Abbildung 4.22: Rechendauer bei Kerndichte Schätzern in Abhängigkeit von der Anzahl der Eingangsdaten für eine Sinusschwingung

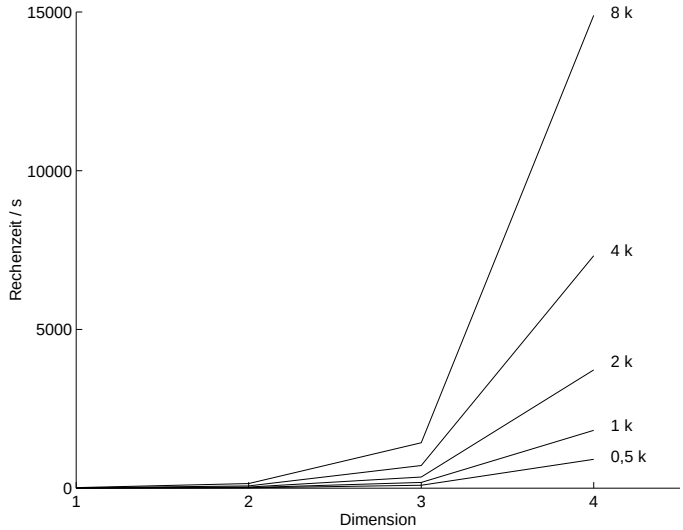


Abbildung 4.23: Rechendauer bei Kerndichte Schätzern in Abhängigkeit von der Dimension für eine Sinusschwingung

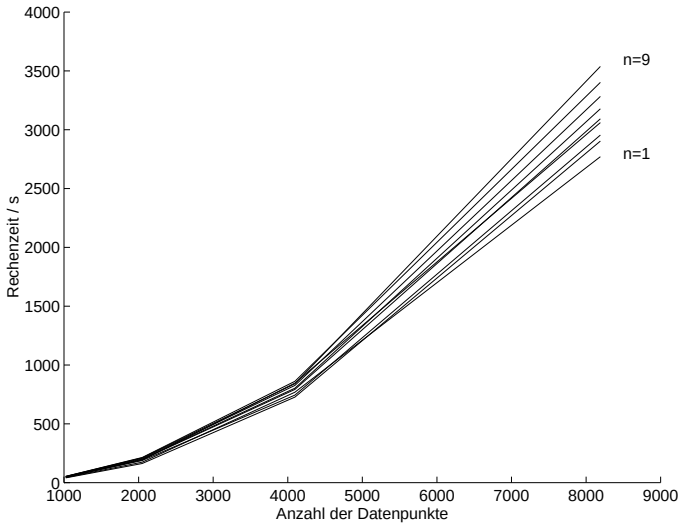


Abbildung 4.24: Rechendauer beim MMI–Alogrithmus in Abhängigkeit von der Anzahl der Eingangsdaten für eine Sinusschwingung

- Die Anzahl der Gitterpunkte pro Koordinate wirkt sich, wie in (4.10) zu sehen ist, mit der Potenz der verwendeten Dimension auf das Laufzeitverhalten des Programmes aus.

Die Struktur der Eingangsdaten geht, wie bereits erwähnt, nicht in die Rechenzeit ein. Ausgenommen davon sind Effekte, die durch die Implementation der Kernfunktion entstehen, diese besitzen aber nur untergeordnete Bedeutung.

4.5.4 MMI–Algorithmus

Die Laufzeit des MMI–Algorithmus kann durch die folgenden Abhängigkeiten charakterisiert werden:

- Anzahl der Datenpunkte N: da der Algorithmus über alle Datenpunkte summiert und zwar jeweils die Beiträge aller anderen Vektoren, verhält sich die Laufzeit proportional zu N^2 (siehe Bild 4.24).

  Kennel hat sich in seinem originalen Algorithmus auf den Epanechnikov Kern beschränkt, der Beiträge nur bis zu einem (normierten) Abstand von 1 liefert. Dadurch wurde es möglich mit Hilfe von KD-Bäumen[6] einen Algorithmus zu entwerfen, dessen Ausführungszeit proportional $N \log N$ ist (siehe Bild 4.25). Im Rahmen dieser Diplomarbeit wurde allerdings versucht, ein möglichst flexibles Programm zu entwerfen. Daher wurden auch Kerne mit unendlicher Ausdehnung berücksichtigt; folglich konnte das schnellere Verfahren nicht eingesetzt werden.

[6] KD-Bäume ermöglichen es, zu einem Datenpunkt die nächsten Nachbarn zu finden.

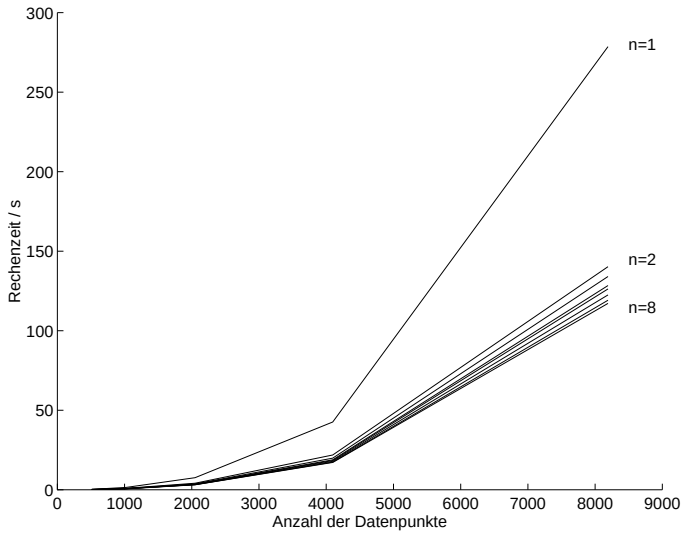


Abbildung 4.25: Rechendauer beim originalen MMI–Alogrithmus in Abhängigkeit von der Anzahl der Eingangsdaten für eine Sinusschwingung

- Dimension n: weil die Summation immer über alle Datenpunkte erfolgt, ist die Berechnung primär unabhängig von der Dimension. Allerdings wächst natürlich die Zeit, die nötig ist, um den Abstand zweier Punkte zu berechnen. Daher resultiert ein Anstieg der gesamten Rechenzeit, der leicht mit der Dimension ansteigt (siehe Bild 4.26).

  Beim originalen Algorithmus von Kennel werden bei der Summation nur die Punkte berücksichtigt, die innerhalb eines gewissen Radius liegen. Da die Abstände zwischen den Punkten mit steigender Dimension zunehmen, werden immer weniger Punkte aufsummiert. Der Algorithmus weist also bei höheren Dimensionen eine geringfügig kürzere Laufzeit auf, wie in Bild 4.27 zu sehen ist.

4.6 Speicherbedarf

4.6.1 Adaptives Histogramm

Der vom Programm benötigte Hauptspeicher wird im wesentlichen von den folgenden zwei Datenstrukturen in Anspruch genommen.

- Den Eingangsdaten. Diese liegen in Form einer verketteten Liste vor. Jeder Vektor wird genau einmal im Hauptspeicher abgelegt.

- Den Randverteilungen. Sie werden in Bäumen gespeichert. Die Verbundwahrscheinlichkeit wird dynamisch errechnet und benötigt daher nur einen vernachlässigbaren Teil des Speichers. Die Randverteilungen hingegen müssen während der gesamten Berechnung zur Verfügung stehen.

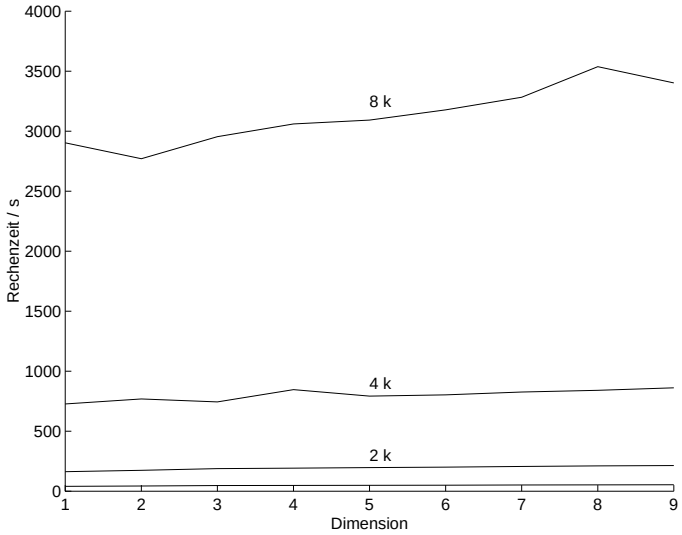


Abbildung 4.26: Rechendauer beim MMI–Algorithmus in Abhängigkeit von der Dimension für eine Sinusschwingung

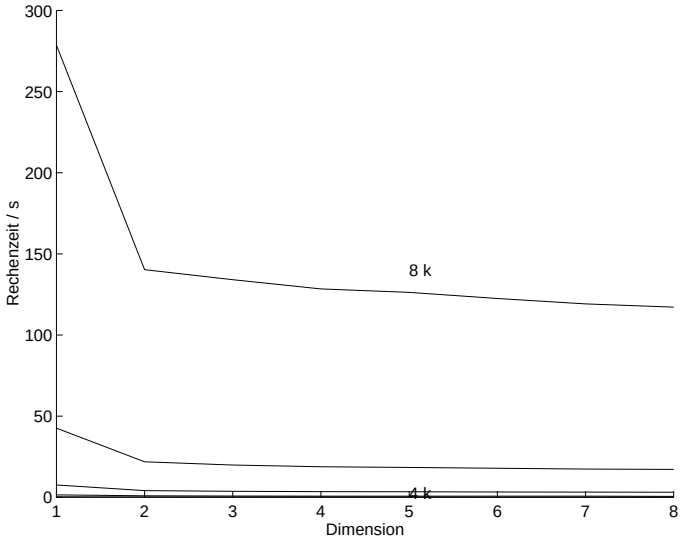


Abbildung 4.27: Rechendauer beim originalen MMI–Algorithmus in Abhängigkeit von der Dimension für eine Sinusschwingung

Über die Abhängigkeit der Speicheranforderungen von der Anzahl an Eingangsdaten, der Dimension sowie der Datenstruktur kann man folgende Aussagen treffen:

- Die Anzahl der Eingangsdaten ist direkt proportional zur Länge der verketteten Liste. Die Baumstruktur wird durch sie nicht beeinflußt. Bei niedrigen Dimensionen wird dadurch ein annähernd lineares Anwachsen des Speicherbedarfes zu beobachten sein. Ab etwa 3 – 4 Dimensionen dominieren jedoch die Baumstrukturen.

- Die verwendete Dimension geht auf zwei verschiedene Arten in den Speicherbedarf ein. In der verketteten Liste steigt der Platzbedarf linear mit der Dimension, da im n –dimensionalen Raum für jeden Punkt n Koordinatenwerte gespeichert werden müssen. Drastischer sind die Auswirkungen auf die Speicheranforderungen der Baumstrukturen, da für jede Unterstruktur 2^n neue Hyperkuben erzeugt werden müssen. Zusätzlich wird die Größe eines Hyperwürfels durch 2^n Zeigervariablen bestimmt, die die Verbindung zu den Unterklassen herstellen.

- Die Form der zu schätzenden Wahrscheinlichkeitsdichte beeinflußt den Speicherbedarf in der selben Art und Weise wie die Laufzeit des Programmes. Je zerklüfteter die Verteilung ist, desto mehr Intervalle müssen verwendet werden, daher steigt der Speicherbedarf äquivalent zur Rechenzeit.

4.6.2 Kerndichte Schätzung

Der benötigte Speicher für dieses Programm setzt sich aus folgenden Komponenten zusammen:

- Einer einfach verkettete Liste mit allen Eingangsvektoren;

- Zwei Speicherbereichen, die die geschätzten Randverteilungsdichten enthalten.

Daher ergeben sich die folgenden Abhängigkeiten:

- Die Anzahl der Datenpunkte verlängert die einfach verkettete Liste proportional. Insgesamt steigt der Speicherbedarf schwächer als linear an.

- Die verwendete Dimension wirkt sich linear auf den Speicherbedarf der einfach verketteten Liste aus (da entsprechend viele Koordinaten abgespeichert werden müssen). Exponentiell dagegen steigt der Bedarf an Speicher zum Berechnen der Randverteilungen. Bei n_1 bzw. n_2 Dimensionen und K Gitterpunkten je Koordinate werden

$$K^{n_1} + K^{n_2} \tag{4.11}$$

  Elemente benötigt, die jeweils eine Gleitkommazahl aufnehmen müssen.

- Ebenfalls aus (4.11) ist ersichtlich, daß auch die Zahl der Gitterstellen sich auf den benötigten Speicherplatz auswirkt, und zwar mit der Potenz der verwendeten Dimension.

4.6.3 MMI–Algorithmus

Der MMI–Algorithmus arbeitet mit einer Liste aller Eingangsvektoren (im originalen Algorithmus wird an Stelle der Liste ein KD-Baum verwendet, der Speicherbedarf ändert sich dadurch allerdings nicht). Jeder Vektor enthält neben seinen Koordinaten auch noch drei Gleitkommazahlen, in denen die summierten Beiträge der Kernfunktionen der Nachbarpunkte abgespeichert werden.

Der Speicherbedarf verhält sich somit linear zur Anzahl der Eingangsdaten und zur verwendeten Dimension (da entsprechend viele Koordinaten abgespeichert werden müssen).

Kapitel 5

Programme

5.1 Transinformation

5.1.1 Adaptives Histogramm

Der Aufruf des Programmes erfolgt mit

```
transinfo -p### -f## -s## -n## -t## [-a##] [-b##] [-k##] -q### [-F##]
[-S##] [-N##] [-T##] [-A##] [-B##] [-K##] [-l##]
```

Dabei können mit den Kommandozeilenoptionen die folgenden Parameter eingestellt werden.

`-p` Name der Datei, die die Abtastwerte des Signals P enthält. Dieser Parameter muß angegeben werden.

`-q` Name der Datei, die die Abtastwerte des Signals Q enthält. Dieser Parameter muß angegeben werden.

`-f` Dateityp für die erste Eingangsdatei. Diese Option kann im einzelnen die Werte

- `1` Textdatei: Jeder Abtastwert steht in einer eigenen Zeile.
- `2` WAV-Datei: Der Aufbau der Datei folgt den Richtlinien für Windows Klangdateien.
- `3` Binärdatei: Die Daten sind im Binärformat angegeben. Die Wortlänge kann mit -s eingestellt werden. Die Daten müssen in der Anordnung Low-Byte / High-Byte vorliegen.

annehmen. Dieser Parameter muß angegeben werden.

`-F` Dateityp für die zweite Eingangsdatei. Die möglichen Werte für diese Option sind identisch mit den Werten für `-f`. Falls dieser Parameter nicht verwendet wird, setzt ihn das Programm automatisch auf den selben Wert, der unter `-f` angegeben wurde.

`-s` Dieser Parameter kann nur zusammen mit `-f 3` verwendet werden und gibt an, wieviele Bytes in der ersten Eingangsdatei einen Abtastwert beschreiben. Da programmintern der Typ long int verwendet wird, ist auf den meisten Systemen ein Maximalwert von 4 sinnvoll. Wird dieser Parameter nicht angegeben, setzt ihn das Programm automatisch auf 1.

`-S` Dieser Parameter kann nur zusammen mit `-F 3` verwendet werden und gibt an, wieviele Bytes in der zweiten Eingangsdatei einen Abtastwert beschreiben. Der Vorgabewert ist 1.

`-n` Rekonstruktionsdimension für die Daten aus der ersten Eingangsdatei. Dieser Parameter muß angegeben werden.

`-N` Rekonstruktionsdimension für die Daten aus der zweiten Eingangsdatei. Wird dieser Parameter nicht angegeben, so setzt ihn das Programm auf den selben Wert, der mit `-n` bestimmt wurde.

`-t` Verzögerungszeit für die Erzeugung von Vektoren aus der ersten Eingangsdatei. Dieser Parameter muß angegeben werden.

`-T` Verzögerungszeit für die Erzeugung von Vektoren aus der zweiten Eingangsdatei. Wird dieser Parameter nicht angegeben, so setzt ihn das Programm auf den selben Wert, der mit `-t` bestimmt wurde.

`-a` Offset, der beim Einlesen von der ersten Eingangsdatei angewendet wird. Mit `-a 100` wird zum Beispiel festgelegt, daß die ersten 100 Werte der mit `-p` bestimmten Datei übersprungen werden. Der Vorgabewert für diese Option ist 0.

`-b` ist nur wirksam, wenn `-k` verwendet wird. Damit wird der maximale Offset, der bei der ersten Datei wirksam werden soll, festgelegt. Mit der Angabe `-b 500` wird zum Beispiel bestimmt, daß das Programm beendet wird, sobald der Offset der ersten Datei grösser als 500 ist.

`-A` Offset, der beim Einlesen von der zweiten Eingangsdatei angewendet wird. Der Vorgabewert für diese Option ist 0.

`-B` ist nur wirksam, wenn `-K` verwendet wird. Damit wird der maximale Offset, der bei der zweiten Datei wirksam werden soll, festgelegt.

`-k` aktiviert den Mehrschrittmodus. Nachdem die Transinformation mit den durch `-a` und `-A` bestimmten Offsets berechnet ist, erhöht das Programm den Offset der ersten Eingangsdatei um den mit `-k` festgelegten Wert und wiederholt die Berechnung. Dies geschieht so lange, bis der Offset größer ist, als der durch `-b` festgelegte Wert. Die Voreinstellung für diesen Parameter ist 0, damit wird der Mehrschrittmodus für die erste Eingangsdatei deaktiviert.

-K funktioniert analog zu `-k`. Der Offset der zweiten Datei wird, ausgehend von `-A`, solange um den durch -K spezifizierten Wert erhöht, bis der durch `-B` festgelegte

Wert überschritten wird. Der Vorgabewert für diesen Parameter ist 0, damit wird der Mehrschrittmodus für die zweite Eingangsdatei ausgeschaltet.

-l bestimmt die Anzahl von Datenpunkten, die vom Programm verwendet wird, falls die Eingangsdateien entsprechend viele Daten enthalten. Der Vorgabewert für diese Option ist 0, damit werden alle Abtastwerte gelesen, die in den Eingangsdateien enthalten sind.

5.1.2 Kerndichte Schätzung

Der Aufruf des Programmes erfolgt mit

```
kdtransinfo -p### -f## -s## -n## -t## [-a##] [-b##] [-k##] -q### [-F##]
[-S##] [-N##] [-T##] [-A##] [-B##] [-K##] [-w##] [-e##] [-l##]
```

Die Kommandozeilenparameter haben dieselbe Bedeutung wie in 5.1.1, zusätzlich können noch folgende Parameter eingestellt werden:

`-w` Bestimmt die Kernbreite des verwendeten Kernes. Mit einer hohen Kernbreite wird die Verteilungsdichte verschmierter, dafür reichen bereits weniger Datenpunkte für eine sinnvolle Berechnung aus. Im Gegensatz dazu wird die Schätzfunktion bei kleiner Kernbreite exakter, allerdings werden dazu mehr Datenpunkte benötigt.

Wird dieser Parameter nicht angegeben, so verwendet das Programm eine Kernbreite von 0.1.

`-e` Wählt eine Kernfunktion aus. Im Augenblick stehen die folgenden beiden Kerne zur Auswahl:

`1` Epanechnikov Kern

`2` Gauss Kern

Eine Ergänzung mit weiteren Kernen ist im Programmcode vorgesehen.

5.1.3 Der MMI–Algorithmus

Der Aufruf des Programmes erfolgt mit

```
kdtransinfo -p### -f## -s## -n## -t## [-a##] [-b##] [-k##] -q### [-F##]
[-S##] [-N##] [-T##] [-A##] [-B##] [-K##] [-i##] [-w##] [-e##] [-l##]
[-d]
```

Die Kommandozeilenparameter haben dieselbe Bedeutung wie in 5.1.1, zusätzlich können noch folgende Parameter eingestellt werden:

`-i` Gibt die Anzahl der Iterationen an, die vom Programm durchgeführt werden sollen. Bei allen Iterationen außer der ersten wird eine Adaption der Kernbreiten vorgenommen.

Wird dieser Parameter nicht angegeben, so wird 1 Iteration durchgeführt.

`-w` Bestimmt die Kernbreite des verwendeten Kernes. Mit einer hohen Kernbreite wird die Verteilungsdichte verschmierter, dafür reichen bereits weniger Datenpunkte für eine sinnvolle Berechnung aus. Im Gegensatz dazu wird die Schätzfunktion bei kleiner Kernbreite exakter, allerdings werden dazu mehr Datenpunkte benötigt.

Wird dieser Parameter nicht angegeben, so verwendet das Programm eine Kernbreite von 0.1.

`-e` Wählt eine Kernfunktion aus. Im Augenblick stehen die folgenden beiden Kerne zur Auswahl:

`1` Epanechnikov Kern

`2` Gauss Kern

Eine Ergänzung mit weiteren Kernen ist im Programmcode vorgesehen.

`-d` Erlaubt unterschiedliche Adaption der Kernbreiten für die Verbundwahrscheinlichkeitsdichte und die Randverteilungen. Dieser Parameter ist nur im Zusammenhang mit `-i` sinnvoll, da erst ab der zweiten Iteration eine Adaption der Kernbreiten erfolgt.

Wird dieser Parameter nicht angegeben, so wird die Kernbreite für einen Datenpunkt sowohl für die Verbundwahrscheinlichkeit als auch für die Randverteilungen gleich gewählt.

5.1.4 Ausgabe

Die Ausgabe erfolgt bei allen Algorithmen zur Berechnung der Transinformation im selben Format. Ein typischer Programmlauf könnte beispielsweise das folgende Resultat liefern:

```
Mutual information calculation
Written in 1995 by Stefan Froehlich
Input files: P = data/sin11-0.bin, n = 1, T = 11
             Q = data/sin11-0.bin, n = 3, T = 11
5000 samples used.
Offset P: 0,   Offset Q: 11,   I(A;B) = 0.409315     (2.47 sec.)
Offset P: 0,   Offset Q: 12,   I(A;B) = 0.863262     (2.89 sec.)
Offset P: 0,   Offset Q: 13,   I(A;B) = 0.757739     (2.89 sec.)
Offset P: 0,   Offset Q: 14,   I(A;B) = 0.892421     (2.81 sec.)
Offset P: 0,   Offset Q: 15,   I(A;B) = 0.99274      (2.93 sec.)
Offset P: 0,   Offset Q: 16,   I(A;B) = 0.792202     (2.76 sec.)
Offset P: 0,   Offset Q: 17,   I(A;B) = 0.694956     (2.7 sec.)
Offset P: 0,   Offset Q: 18,   I(A;B) = 0.734847     (2.67 sec.)
Offset P: 0,   Offset Q: 19,   I(A;B) = 0.581344     (2.6 sec.)
Offset P: 0,   Offset Q: 20,   I(A;B) = 0.424659     (2.61 sec.)
Offset P: 0,   Offset Q: 21,   I(A;B) = 0.691193     (2.74 sec.)
Offset P: 0,   Offset Q: 22,   I(A;B) = 0.865399     (2.9 sec.)
```

```
Offset P: 0,   Offset Q: 23,   I(A;B) = 0.492546     (2.59 sec.)
Offset P: 0,   Offset Q: 24,   I(A;B) = 0.946984     (2.83 sec.)
Offset P: 0,   Offset Q: 25,   I(A;B) = 0.786832     (2.72 sec.)
Offset P: 0,   Offset Q: 26,   I(A;B) = 0.786874     (2.72 sec.)
Offset P: 0,   Offset Q: 27,   I(A;B) = 1.29419      (3.07 sec.)
Offset P: 0,   Offset Q: 28,   I(A;B) = 0.821607     (2.76 sec.)
```

In den Kopfzeilen der Ausgabe stehen alle wichtigen Parameter, die über die Kommandozeile angegeben wurden. Dabei handelt es sich um die Dateinamen für die Eingangsdaten, um die verwendeten Dimensionen und um die Verzögerung für die Konstruktion der Vektoren. Außerdem wird die Maximalzahl[1] der verwendeten Vektoren angegeben.

Anschließend erfolgt die Ausgabe der Rechenergebnisse. Wurde das Programm nicht mit `-k` oder `-K` im Mehrschrittmodus gestartet, so besteht dieser Teil nur aus einer einzigen Zeile. Im Mehrschrittmodus wird für jeden Schritt eine eigene Zeile ausgegeben. In der Ergebniszeile sind die beiden jeweiligen Offsets angegeben, die beim Einlesen der Datei berücksichtigt wurden. Im Anschluß daran befindet sich die errechnete Transinformation. Abschließend wird in Klammern die benötigte Rechenzeit[2] angegeben.

5.2 Kullback Leibler Distanz

5.2.1 Aufruf

Der Aufruf des Programmes erfolgt mit

```
kullback -p### -f## -s## -n## -t## [-a##] [-b##] [-k##] -q### [-F##]
[-S##] [-T##] [-A##] [-B##] [-K##] [-l##]
```

Dabei können mit den Kommandozeilenoptionen die folgenden Parameter eingestellt werden.

`-p` Name der Datei, die die Abtastwerte des Signals P enthält. Dieser Parameter muß angegeben werden.

`-q` Name der Datei, die die Abtastwerte des Signals Q enthält. Dieser Parameter muß angegeben werden.

`-f` Dateityp für die erste Eingangsdatei. Diese Option kann im einzelnen die Werte

`1` Textdatei: Jeder Abtastwert steht in einer eigenen Zeile.

`2` WAV-Datei: Der Aufbau der Datei folgt den Richtlinien für Windows Klangdateien.

[1] Diese kann unterschritten werden, wenn eine der beiden Eingangsdatein weniger Daten enthält.

[2] Dabei handelt es sich nur um die tatsächlich verbrauchte CPU-Zeit. Das Programm kann also durchaus länger arbeiten, falls auf demselben Rechner noch andere Programme ausgeführt werden.

`3` Binärdatei: Die Daten sind im Binärformat angegeben. Die Wortlänge kann mit -s eingestellt werden. Die Daten müssen in der Anordnung Low-Byte / High-Byte vorliegen.

annehmen. Dieser Parameter muß angegeben werden.

`-F` Dateityp für die zweite Eingangsdatei. Die möglichen Werte für diese Option sind identisch mit den Werten für `-f`. Falls dieser Parameter nicht verwendet wird, setzt ihn das Programm automatisch auf den selben Wert, der unter `-f` angegeben wurde.

`-s` Dieser Parameter kann nur zusammen mit `-f 3` verwendet werden und gibt an, wieviele Bytes in der ersten Eingangsdatei einen Abtastwert beschreiben. Da programmintern der Typ long int verwendet wird, ist auf den meisten Systemen ein Maximalwert von 4 sinnvoll. Wird dieser Parameter nicht angegeben, setzt ihn das Programm automatisch auf 1.

`-S` Dieser Parameter kann nur zusammen mit `-F 3` verwendet werden und gibt an, wieviele Bytes in der zweiten Eingangsdatei einen Abtastwert beschreiben. Der Vorgabewert ist 1.

`-n` Rekonstruktionsdimension. Bei der Berechnung der Kullback Leibler Distanz kann nur eine Dimension angegeben werden, die für beide Eingangsdateien verwendet wird.

`-t` Verzögerungszeit für die Erzeugung von Vektoren aus der ersten Eingangsdatei. Dieser Parameter muß angegeben werden.

`-T` Verzögerungszeit für die Erzeugung von Vektoren aus der zweiten Eingangsdatei. Wird dieser Parameter nicht angegeben, so setzt ihn das Programm auf den selben Wert, der mit `-t` bestimmt wurde.

`-a` Offset, der beim Einlesen von der ersten Eingangsdatei angewendet wird. Mit `-a 100` wird zum Beispiel festgelegt, daß die ersten 100 Werte der mit `-p` bestimmten Datei übersprungen werden. Der Vorgabewert für diese Option ist 0.

`-b` ist nur wirksam, wenn `-k` verwendet wird. Damit wird der maximale Offset, der bei der ersten Datei wirksam werden soll, festgelegt. Mit der Angabe `-b 500` wird zum Beispiel bestimmt, daß das Programm beendet wird, sobald der Offset der ersten Datei grösser als 500 ist.

`-A` Offset, der beim Einlesen von der zweiten Eingangsdatei angewendet wird. Der Vorgabewert für diese Option ist 0.

`-B` ist nur wirksam, wenn `-K` verwendet wird. Damit wird der maximale Offset, der bei der zweiten Datei wirksam werden soll, festgelegt.

`-k` aktiviert den Mehrschrittmodus. Nachdem die Transinformation mit den durch `-a` und `-A` bestimmten Offsets berechnet ist, erhöht das Programm den Offset der ersten Eingangsdatei um den mit `-k` festgelegten Wert und wiederholt die Berechnung. Dies geschieht so lange, bis der Offset größer ist, als der durch `-b` festgelegte Wert. Die Voreinstellung für diesen Parameter ist 0, damit wird der Mehrschrittmodus für die erste Eingangsdatei deaktiviert.

`-K` funktioniert analog zu `-k`. Der Offset der zweiten Datei wird, ausgehend von `-A`, solange um den durch -K spezifizierten Wert erhöht, bis der durch `-B` festgelegte Wert überschritten wird. Der Vorgabewert für diesen Parameter ist 0, damit wird der Mehrschrittmodus für die zweite Eingangsdatei ausgeschaltet.

`-l` bestimmt die Anzahl von Datenpunkten, die vom Programm für das Signal $p(x)$ verwendet wird, falls die Eingangsdatei entsprechend viele Daten enthält. Der Vorgabewert für diese Option ist 0, damit werden alle Abtastwerte gelesen, die in der ersten Eingangsdatei enthalten sind.

`-L` bestimmt die Anzahl von Datenpunkten, die vom Programm für das Signal $q(x)$ verwendet wird, falls die Eingangsdatei entsprechend viele Daten enthält. Der Vorgabewert für diese Option ist 0, damit werden alle Abtastwerte gelesen, die in der zweiten Eingangsdatei enthalten sind.

5.2.2 Ausgabe

Ein typischer Programmlauf zur Berechnung der Kullback Leibler Distanz könnte beispielsweise die folgende Ausgabe liefern:

```
Kullback-Leibler distance calculation
Written in 1995 by Stefan Froehlich
Input files: P = data/sin11-0.dat, T = 11, length = 5000
             Q = data/sin11-0.dat, T = 11, length = 5000
Dimension = 1
Offset P: 0,   Offset Q: 11,   D(A|B) = 0.126464     (4.32 sec.)
Offset P: 0,   Offset Q: 12,   D(A|B) = 0.120518     (4.43 sec.)
Offset P: 0,   Offset Q: 13,   D(A|B) = 0.119269     (4.48 sec.)
Offset P: 0,   Offset Q: 14,   D(A|B) = 0.11409      (4.46 sec.)
Offset P: 0,   Offset Q: 15,   D(A|B) = 0.125612     (4.35 sec.)
Offset P: 0,   Offset Q: 16,   D(A|B) = 0.0956377    (4.32 sec.)
Offset P: 0,   Offset Q: 17,   D(A|B) = 0.0756519    (4.31 sec.)
Offset P: 0,   Offset Q: 18,   D(A|B) = 0.0920888    (4.19 sec.)
Offset P: 0,   Offset Q: 19,   D(A|B) = 0.0875948    (4.18 sec.)
Offset P: 0,   Offset Q: 20,   D(A|B) = 0.0761001    (4.16 sec.)
Offset P: 0,   Offset Q: 21,   D(A|B) = 0.0796012    (4.16 sec.)
Offset P: 0,   Offset Q: 22,   D(A|B) = 0.109635     (4.22 sec.)
Offset P: 0,   Offset Q: 23,   D(A|B) = 0.0994239    (4.18 sec.)
Offset P: 0,   Offset Q: 24,   D(A|B) = 0.0720285    (4.12 sec.)
```

```
Offset P: 0,   Offset Q: 25,   D(A|B) = 0.0619929    (4.14 sec.)
Offset P: 0,   Offset Q: 26,   D(A|B) = 0.121633     (4.31 sec.)
Offset P: 0,   Offset Q: 27,   D(A|B) = 0.121785     (4.26 sec.)
Offset P: 0,   Offset Q: 28,   D(A|B) = 0.147328     (4.27 sec.)
```

In den Kopfzeilen der Ausgabe stehen alle wichtigen Parameter, die über die Kommandozeile angegeben wurden. Dabei handelt es sich um die Dateinamen für die Eingangsdaten, um die verwendete Dimension und um die Verzögerung für die Konstruktion der Vektoren. Außerdem wird die Maximalzahl[3] der verwendeten Vektoren für jede Eingangsdatei angegeben.

Anschließend erfolgt die Ausgabe der Rechenergebnisse. Wurde das Programm nicht mit `-k` oder `-K` im Mehrschrittmodus gestartet, so besteht dieser Teil nur aus einer einzigen Zeile. Im Mehrschrittmodus wird für jeden Schritt eine eigene Zeile ausgegeben. In der Ergebniszeile sind die beiden jeweiligen Offsets angegeben, die beim Einlesen der Datei berücksichtigt wurden. Im Anschluß daran befindet sich die errechnete Kullback Leibler Distanz. Abschließend wird in Klammern die benötigte Rechenzeit[4] angegeben.

5.3 Format der Eingangsdateien

Die Eingangsdateien können zur Zeit in einem von drei möglichen Formaten bereitgestellt werden. Dabei handelt es sich um

- ASCII Textdateien. Das Programm erwartet in jeder Zeile der Eingangsdatei genau eine Zahl. Dabei darf es sich auch um einen Gleitkommawert handeln.

  Da bei Textdateien die Anzahl der Zeilen nicht im voraus bestimmt werden kann, müssen solche Dateien zunächst vom Programm vermessen werden, was bei großen Datenmengen (ab etwa 10^6 Punkten) zu einer zusätzlichen Verzögerung von einigen Sekunden führen kann.

- Binäre Dateien. Hier sind nur ganze Zahlen als Eingangsdaten zulässig. Es darf sich nur um $8k$ Bit breite Werte handeln, wobei k bei den meisten Maschinen auf 4 begrenzt ist (entsprechend dem Datentyp `long int`).

  Die Eingangsdaten müssen im Low Byte / High Byte Format vorliegen.

- WAV Dateien. Diese Dateien entsprechen dem Standard, der für Windows Klangdateien vorgesehen ist. Die genauen Spezifikationen sind aus der entsprechenden Literatur zu entnehmen.

Eine Erweiterung dieser Liste auf neue Formate ist problemlos möglich. Dazu muß lediglich eine neue Unterklasse von der Klasse `Datafile` angelegt werden, wie das in `binfile.h` und `binfile.cc` beispielsweise für Binärdateien getan wurde.

[3] Diese kann unterschritten werden, wenn eine der beiden Eingangsdatein weniger Daten enthält

[4] Dabei handelt es sich nur um die tatsächlich verbrauchte CPU-Zeit. Das Programm kann also durchaus länger arbeiten, falls auf demselben Rechner noch andere Programme ausgeführt werden.

Kapitel 6

Zusammenfassung

In Tabelle 6.1 werden die mit den verschiedenen Algorithmen erzielten Ergebnisse einander gesammelt gegenübergestellt. Für jedes Verfahren wird die Rechengenauigkeit, die Ausführungszeit sowie der Speicherbedarf angegeben.

Es zeigt sich, daß der auf dem adaptiven Histogramm basierende, neu entwickelte Algorithmus eine deutlich höhere Rechengenauigkeit aufweist als die anderen untersuchten Verfahren. Für die Kerndichte Schätzung kann hier kein Wert angegeben werden, da, um eine ausreichende Anzahl an Meßwerten zu erhalten, die Rechenzeit zu groß ist. Untersuchungen im eindimensionalen Fall haben gezeigt, daß der Fehler bei der Kerndichteschätzung die selbe Größenordnung aufweist wie beim adaptiven Histogramm (siehe Kapitel 3.3.3). Die Ergebnisse des MMI–Algorithmus sind nicht verwendbar.

Im Gegensatz dazu stehen die Ergebnisse für das dynamische Verhalten der Programme. Der von H.-P. Bernhard entwickelte FMI–Algorithmus weist eine Laufzeit aus, die sich sowohl linear zur Anzahl der Eingangsdaten als auch linear zur bei der Berechnung verwendeten Dimension verhält. Bei den neuentwickelten Algorithmen hingegen steigt die Rechenzeit exponentiell mit der Dimension an (die Abhängigkeit von der Anzahl der Datenpunkte ist wie beim FMI–Algorithmus linear). Lediglich der MMI–Algorithmus weist hier bessere Werte auf (beispielsweise ist die Laufzeit von der Dimension völlig unabhängig), auf Grund des unzulässig hohen Rechenfehlers ist dies allerdings uninteressant.

Der Speicherbedarf steht bei allen Verfahren in direktem Zusammenhang mit der Ausführungszeit.

Abschließend kann daher festgestellt werden: *Für Berechnungen, bei denen auf hohe Rechengenauigkeit Wert gelegt wird, kann der neu entwickelte Algorithmus basierend auf dem adaptiven Histogramm empfohlen werden. Bei Anwendungen, die größere Rechenfehler zulassen, während die Ausführungszeit kritisch ist, eignen sich die Programme aus [2] besser.*

	FMI–Algor.	Adapt. Hist.	Kerndichte Sch.	MMI–Algor.
Transinformation für SNR = 42 dB	6,89	≈ 7	-	$\approx 4,5$
Laufzeit (Datenpunkte)	linear	linear	linear	$N \log N$
Laufzeit (Dimension)	linear	epxonentiell	exponentiell	-
Speicher (Datenpunkte)	linear	linear	linear	linear
Speicher (Dimension)	linear	exponentiell	exponentiell	linear

Tabelle 6.1: Gegenüberstellung der Ergebnisse

Literaturverzeichnis

[1] Bernhard H.-P.: *Analyse von Sprachsignalen mit Methoden der Chaostheorie.* Diplomarbeit an der TU Wien (1991)

[2] Bernhard H.-P. und Kubin G.: *A fast mutual information calculation algorithm.* Proceedings of the 7th European Signal Processing Conference EUSIPCO'94, Edinburgh (1994)

[3] Bronstein I.N., Semendjajew K.A.: *Taschenbuch der Mathematik.* Teubner, Leipzig (1983)

[4] Crutchfield J.P., Farmer J.D., Packert N.H. und Shaw R.: *Chaos.* Spektrum der Wissenschaften Chaos und Fraktale (1989)

[5] Dirschmid H.J.: *Mathematische Grundlagen der Elektrotechnik.* Vieweg, Braunschweig (1986)

[6] Epanechnikov V.K.: *Non-Parametric Estimation of a Multivariate Propability Density.* Theory Propab. Appl. No. 14, S. 153-158 (1969)

[7] Fraser A.M.: *Information and Entropy in Strange Attractors.* IEEE Transactions on Information Theory, VOL IT - 35, no. 2 (1989)

[8] Hartung J.: *Statistik.* R. Oldenburg Verlag München (1987)

[9] Knapp A.: *Serienuntersuchung von Sprachsignalen mit Methoden der Chaostheorie.* Diplomarbeit an der TU Wien (1994)

[10] Ledermann W.: *Statistics* John Wiley & Sons, Inc. (1984)

[11] Nonner Ch.: *Graphische Verfahren zur Darstellung und Beurteilung von Verteilungen.* Diplomarbeit an der TU Wien (1986)

[12] Scott D.W.: *Multivariate Density Estimation.* John Wiley & Sons, Inc. (1992)

[13] Terrel G.R. und Scott D.W.: *Variable Kernel Density Estimation.* Annual Statistics (1992)

Abbildungsverzeichnis