"Arc Diagrams"
zur Visualisierung von Mustern in multivariaten Zeitreihen

Diplomarbeit

vorgelegt von:	B.Sc. Michael Zornow
Betreuer:	Dr.-Ing. Christian Tominski
Gutachter:	Prof. Dr.-Ing. habil. Heidrun Schumann
	Prof. Dr.-Ing. habil. Peter Forbrig
Abgabedatum:	28.02.2007

Universität Rostock

Fakultät für Informatik und Elektrotechnik

Zusammenfassung

Die expressive Darstellung von zeitabhängigen Daten ist eine der wichtigsten Aufgaben in der Informationsvisualisierung. Daher werden seit langer Zeit immer neue Visualisierungstechniken für solche Daten entwickelt. Diese beschränken sich jedoch meist auf die Darstellung der vorliegenden Daten, eine Hervorhebung von wiederkehrenden Mustern wird nicht in ausreichendem Maße unterstützt. Doch genau solche Muster sind es, die wesentlich zum Verständnis der in den Daten enthaltenen Informationen beitragen.

Die „Arc Diagram" Technik ist ein Verfahren zur Darstellung von wiederkehrenden Mustern in Zeichenketten, welches auch für 1-dimensionale Zeichenreihen eingesetzt werden kann. Ziel dieser Arbeit ist es, die „Arc Diagram" Technik so zu erweitern, dass auch Muster in multivariaten Zeitreihen visualisiert werden können. Dazu erfolgt eine Klassifikation, Einordnung und Bewertung der „Arc Diagrams", um zielgerichtet eine Auswahl neuer Konzepte zu ermöglichen. Die entwickelten Konzepte dienen zum Beispiel der Beschleunigung und der Steigerung der Flexibilität der Mustersuche sowie der Verbesserung der Darstellung der „Arc Diagrams" speziell für multivariate Zeitreihen. Dieses und weitere neu entwickelte Konzepte werden in einem Visualisierungstool umgesetzt, welche die Analyse von multivariaten Zeitreihen mit Hilfe der Arc Diagrams zulassen.

Abstract

The effective display of time-dependent data is one of the major tasks in information visualization. There are many expressive techniques known, which however are often limited to the pure visualization of data. To gain insight into the data, support for higher order visualization tasks, like the detection of patterns, is required. However, an explicit indication of repeated patterns within the data is only rarely considered.

"Arc Diagrams" is a technique to visualize repeated patterns in strings (i.e., 1-dimensional data). The goal of this thesis is to extend the "Arc Diagrams" in order to make them applicable to visualize multivariate time-dependent data. Addressing this challenge, "Arc Diagrams" are classified and assessed in order to support the selection of eligible new concepts. Several new concepts, like an improved pattern search strategy and enhanced visual representations, are implemented in an interactive application capable of visualizing multivariate time-series data by means "Arc Diagrams".

CR-Klassifikation

I.3., I.3.6., I.3.8., I.5.

Key Words

Information Visualization, Visual Exploration, Pattern Recognition, Multivariate Data, Time Series

Inhaltsverzeichnis

Kapitel 1

Einleitung und Motivation

Mit Hilfe seiner Sinnesorgane besitzt der Mensch die Fähigkeit, mehrdimensionale Daten auszuwerten. Bedingt durch den technologischen Fortschritt, sieht sich der Mensch jedoch mit einer immer größer werdenden Menge an Daten, die es auszuwerten gilt, konfrontiert. Anwendungen sammeln automatisiert Daten, Sensoren zeichnen in Echtzeit die Daten vieler unterschiedlicher Parameter auf. Aufgrund des Umfangs dieser Daten ist die menschliche Fähigkeit zur mehrdimensionalen Auswertung ohne wissenschaftliche Hilfsmittel längst überfordert. Es müssen Wege gefunden werden, den Menschen zu unterstützen.

Einer dieser Wege ist der Einsatz wissenschaftlicher Methoden zur Extraktion von Informationen und verborgenem Wissen. Gerade zur Analyse großer Datenmengen ist die Anwendung vielfältiger Methoden unerlässlich. Die Verfahren zur Mustererkennung nehmen dabei einen großen Raum ein. Ein solches Verfahren zur visuellen Mustererkennung sind die Arc Diagrams.

Arc Diagrams durchsuchen Datenmengen zunächst nach Mustern und stellen diese Muster anschließend bildlich dar [Wa02]. Diese bildliche Darstellung wird auch als Visualisierung bezeichnet. Sie hat den Vorteil, dass Menschen besser in der Lage sind die relevanten Informationen aufzunehmen, als dies etwa durch eine textliche Beschreibung möglich wäre.

Bisher ist der Einsatz der Arc Diagrams zur visuellen Mustererkennung auf 1-dimensionale Daten, also Zeichenketten, beschränkt. Viele Datenmengen liegen jedoch mehrdimensional vor. Zur visuellen Mustererkennung mehrdimensionaler Daten existieren bislang wenige Verfahren. Ziel dieser Arbeit ist es daher, die Arc Diagrams an ein mehrdimensionales Umfeld anzupassen. Mit Hilfe der Arc Diagrams sollen multivariate Zeitreihen visualisiert werden.

Dazu führt das zweite Kapitel zunächst in das Gebiet der Visualisierung ein und es werden wichtige Begriffe definiert. Das dritte Kapitel dient der exemplarischen Vorstellung einiger Verfahren zur Visualisierung von Mustern. Ein Vergleich dieser Verfahren zeigt die zu lösenden Probleme

speziell der Arc Diagrams für die Visualisierung von Mustern in multivariaten Zeitreihen.

Ein Problem dabei ist es, dass multivariate Zeitreihen gegenüber Zeichenfolgen in der Regel eine wesentlich erhöhte Datenmenge aufweisen. Die Suche und auch die Darstellung dieser höheren Datenmenge sind mit einer Reihe von Herausforderungen verbunden. Zum Beispiel können automatische Voreinstellungen und Parameter durch den Umfang und die Vielfalt der multivariaten Datenmenge nicht immer optimal sein. Deshalb muss der Nutzer einen größeren Einfluss auf den Such- und Darstellungsprozess erhalten. Im vierten Kapitel wird dazu die Integration verschiedener Interaktionstechniken in die Arc Diagrams vorgeschlagen.

Eine weitere Herausforderung ist es, dass sich Zeichenfolgen zumeist auf kleine Alphabete beschränken. Der Text dieser Einleitung beschränkt sich z.B. auf etwa 30 unterschiedliche Zeichen. In Zeitreihen hingegen treten häufig sehr viel mehr unterschiedliche Ausprägungen auf. Im vierten Kapitel wird deshalb eine flexiblere Mustersuche vorgestellt. Diese soll auch Muster in Zeitreihen mit großem Wertebereichsumfang finden.

Der Algorithmus zur Mustersuche der Arc Diagrams ist für die Suche in einer Zeichenfolge konzipiert. In dieser Arbeit sollen jedoch Muster in mehr als einer einzigen Zeichenfolge bzw. Merkmalreihe gefunden werden. Dafür ist die Mustersuche der Arc Diagrams nicht ausgelegt und infolgedessen zu langsam. Im vierten Kapitel wird aus diesem Grund ein Beschleunigungskonzept der Mustersuche vorgestellt.

Neben Mustern bezüglich einer Merkmalreihe, können in multivariaten Zeitreihen Zusammenhänge über einzelne Merkmalreihen hinweg existieren. Diese mehrdimensionalen Muster werden bislang nicht erkannt, da sie in einzelnen Zeichenfolgen bzw. 1-dimensionalen Zeitreihen nicht auftreten können. Im vierten Kapitel sollen intuitive Ideen vorgestellt werden, welche sowohl die Suche nach solchen mehrdimensionalen Mustern, als auch deren Darstellung ermöglichen.

Im fünften Kapitel soll die Implementierung des vorgestellten Konzepts beschrieben werden. Die Implementierung greift die wesentlichen Schwerpunkte und Problemlösungen des vierten Kapitels auf. Auf diese Weise dient das fünfte Kapitel als Proof-Of-Concept. Das sechste Kapitel fasst die wichtigsten Erkenntnisse der Arbeit zusammen. Es gibt abschließend einen Ausblick auf mögliche Erweiterungen des erarbeiteten Konzepts.

Kapitel 2

Grundlagen

In diesem Kapitel sollen die für diese Arbeit fundamentalen Begriffe eingeführt werden. Zunächst wird die Visualisierung vorgestellt. Die Einführung der Begriffe in diesem Umfeld stützt sich vor allem auf die Aussagen in [SM00]. Anschließend wird ein kurzer Einblick in die Einflussfaktoren einer Visualisierung gegeben. Es werden sowohl die Repräsentation der vorliegenden Daten als auch die Bearbeitungsziele klassifiziert.

2.1 Der Visualisierungsprozess

Seit Jahrhunderten beschäftigen sich Menschen mit der bildlichen Darstellung von Informationen. Der Grund dafür ist, dass der visuelle Sinneseindruck beim Menschen evolutionsbedingt am besten ausgeprägt ist. Die vergleichsweise schnelle und intuitive visuelle Wahrnehmungs- und Verarbeitungsfähigkeit unseres Gehirns wird beim Prozess der Visualisierung ausgenutzt.

2.1.1 Allgemeine Betrachtungen

Auch in den Computerwissenschaften ist der Prozess der Erzeugung von graphischen Darstellungen aus abstrakten Daten oder Zusammenhängen, also der Visualisierung, ein traditionelles Aufgabengebiet. Das allgemeine Ziel einer Visualisierung kann nach [SM00] wie folgt formuliert werden:

> *„Ziel der Visualisierung ist es, abstrakte Daten und Informationen so zu präsentieren, dass ihre relevanten Charakteristika intuitiv erfasst werden können.*
>
> *Die Bilder sollen so aufgebaut sein, dass der Betrachter in der Lage ist, nicht nur zu sehen, sondern auch zu erkennen, zu verstehen und zu bewerten."*

Neben dem Prozess der geeigneten Abbildung auf graphische Primitive, welcher anschließend noch diskutiert werden soll, kann man folgende Anforderungen an eine Darstellung formulieren. Sie soll:

- expressiv,
- effektiv
- und angemessen

sein.

Expressiv wird eine Darstellung genannt, wenn sie genau die in den Daten enthaltenen Informationen anzeigt. Dies bedeutet implizit, dass keine Informationen suggeriert werden dürfen, die in den Daten nicht enthalten sind. Wenn eine Visualisierung darüber hinaus *„…die (visuellen) Fähigkeiten des Betrachters und die charakteristischen Eigenschaften des Ausgabegerätes unter Berücksichtigung der Zielsetzung und des Anwendungskontextes optimal ausnutzt…*" [SM00], ist die Forderung nach der Effektivität erfüllt. Die Angemessenheit schließlich macht eine Aussage über Kosten und Nutzen einer Darstellung. Liegt ein ausgewogenes Verhältnis zwischen beiden vor, dann spricht man von einer angemessenen Visualisierung.

2.1.2 Die Visualisierungspipeline

Bereits kurz erwähnt wurde, dass bei der Bildgenerierung ein Prozess der Abbildung von Daten auf graphische Primitive durchlaufen wird. Dieser Prozess wird als Visualisierungspipeline bezeichnet. Er kann in mehrere Stufen eingeteilt werden, welche schrittweise durchlaufen werden. Im Einzelnen sind dies die Abschnitte Filtering, Mapping und Rendering (vgl. Abb. 2-1).

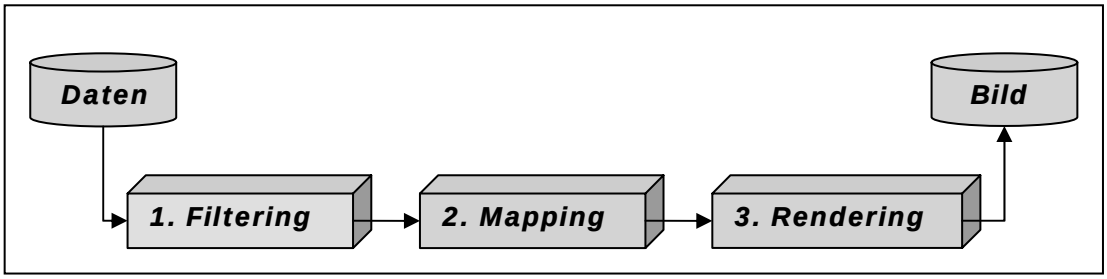

Abbildung 2-1: Die Stufen der Visualisierungspipeline

Das Filtering realisiert eine Daten-zu-Daten Abbildung. Ausgangspunkt dabei sind die in einer Anwendung erhobenen Daten, häufig als Rohdaten bezeichnet. Diese werden für die nachfolgenden Visualisierungsschritte aufbereitet. Hierzu gehört z.B. die Vervollständigung oder Reduzierung der Datenmenge oder das Ableiten bestimmter Kenngrößen (Mittelwerte, Extrema etc.). Als Ergebnis dieses ersten Schrittes liegen die so

genannten aufbereiteten Daten vor. Diese werden an die nächste Stufe der Pipeline übergeben – das Mapping.

Das Mapping ist der wichtigste Schritt des Visualisierungsprozesses. Es wird an dieser Stelle eine Daten-zu-Geometrie Abbildung vorgenommen, bei der die Daten auf geometrische Primitive und ihre graphischen Attribute (Farbe, Helligkeit, Position etc.) abgebildet werden. Das Mapping hat einen entscheidenden Einfluss auf die spätere visuelle Repräsentation der Daten und damit auf die eingeführten Gütekriterien Expressivität, Effektivität und Angemessenheit. Eine ausführlichere Diskussion erfolgt z.B. in [SM00].

Im letzten Schritt der Pipeline geht es darum, die ermittelten Geometriedaten in Bilder zu überführen. Es erfolgt eine Geometrie-zu-Bild Abbildung, die aus visuellen Attributen Bilder erzeugt. In diesem Rendering-Schritt kommen bekannte Bildgenerierungsverfahren zum Einsatz, dessen Erläuterung an der Stelle zu weit gehen würde. Für den interessierten Leser sei auf [ESK97] verwiesen – hier wird der gesamte Prozess der Bildgenerierung ausführlich behandelt.

2.1.3 Allgemeine Visualisierungskonzepte

Im Folgenden sollen einige Visualisierungskonzepte skizziert werden, die für den weiteren Verlauf der Arbeit von Bedeutung sind. Für eine ausführlichere Diskussion dieser Konzepte sei der interessierte Leser z.B. auf [Sp06] oder [Sh96] verwiesen.

Übersicht und Detail

In der Praxis der Visualisierung entstehen, durch sehr große Datenmengen, häufig Bilder, deren Darstellung mehr Platz benötigt, als auf einem graphischen Ausgabegerät zur Verfügung steht. Selbst vergleichsweise große Displays sind nicht in der Lage die rasant wachsenden Datenmengen, welche von immer leistungsstärkeren Sensoren geliefert werden, im Ganzen darzustellen. Eine Skalierung, speziell eine Verkleinerung, würde es erlauben eine Darstellung vollständig zu zeigen. Aufgrund der dazu oft unzureichenden Auflösung von graphischen Anzeigegeräten gehen dabei feine Details jedoch verloren. Ein weiteres Problem offenbart sich, wenn die Auflösung des Anzeigegerätes ausreichen würde: Auch das visuelle System des Menschen hat nur ein bestimmtes Auflösungsvermögen. Die Darstellung würde unter Umständen so klein, dass sie vom Nutzer nicht mehr erkannt werden kann. In Folge dessen kann man zwei Anforderungen an eine Visualisierung formulieren:

1. Die Gesamtheit der Daten sollte in geeigneter Weise dargestellt werden, um einem Nutzer einen Überblick über die Daten zu erlauben.

2. Es muss möglich sein, Details darzustellen, um dem Nutzer auch Feinheiten im visuellen Analyseprozess anzuzeigen.

Das Problem der Darstellung zu vieler Daten auf einer zu kleinen Displayfläche nennt [Sp06] „presentation problem". Eine mögliche Lösung dieses Problems ist das grundlegende Übersicht und Detail Konzept. Hier werden sowohl eine detailreiche Darstellung der Daten als auch eine Übersichtsdarstellung angeboten. Die Trennung der beiden Darstellungen kann entweder temporal oder räumlich sein. Bei der temporalen Trennung werden zeitlich alternierend die beiden Darstellungen angezeigt. Abbildung 2-2 zeigt die räumliche Trennung. Es werden zwei verschiedene Bereiche des Anzeigegerätes benutzt, um beide Darstellungen anzuzeigen: Die Detaildarstellung erstreckt sich über den gesamten Bildschirm und wird in der rechten unteren Ecke von der Übersichtsdarstellung vollständig verdeckt.

Abbildung 2-2: Übersicht und Detail Konzept eines Routenplaners mit räumlicher Trennung, [www.map24.de]

Interessant ist es sicherlich zu untersuchen, inwieweit das vorgestellte Übersicht und Detail Konzept für die Darstellung von multivariaten Zeitreihen genutzt werden kann, um die Arc Diagram Technik bei der Darstellung zu unterstützen. Deshalb wird dieser Ansatzpunkt im vierten Kapitel noch einmal aufgegriffen.

Scrolling und Panning

Nach [Sp06] ist das Scrolling eine sehr verbreitete Lösung, um dem zuvor beschriebenen „presentation problem" zu begegnen. Dem Nutzer wird eine virtuelle Anzeigefläche zur Verfügung gestellt. Mit Hilfe von Bildlaufleisten kann er verschiedene Bereiche der virtuellen Anzeigefläche erreichen, von denen jeweils nur ein Teil auf dem visuellen Ausgabegerät angezeigt wird. Eine dem Scrolling sehr ähnliche Technik ist das Panning.

Der einzige Unterschied zum Scrolling besteht darin, dass das Sichtfenster einer größeren virtuellen Anzeigefläche nicht durch das Verschieben von Bildlaufleisten, sondern durch das „Anfassen und Ziehen" des Bildes selbst, mit der Maus oder ähnlichem, erreicht wird. Das linke Bild in der Abbildung 2-3 verdeutlicht die Möglichkeit der Darstellung durch das Scrolling bzw. Panning. Die unterschiedlichen Sichtfenster 1, 2 und 3 (unten) sollen dabei als drei mögliche Darstellungen der Gesamtdarstellung (oben) verstanden werden.

Der Vorteil des Scrolling und des Panning, verschiedene Ausdehnungen einer Darstellung erreichen zu können, welche nicht auf einmal in ein Darstellungsfenster passen, ist leicht einzusehen. Dennoch bringen sie eine Reihe von Nachteilen mit sich. So ist es z.B. vergleichsweise schwer, sich in einer Darstellung zurechtzufinden, da immer nur ein kleiner Teil der gesamten Darstellung gleichzeitig angezeigt wird. Im vierten Kapitel wird die Frage aufgeworfen werden, ob es trotzdem lohnend ist, das Scrolling und Panning, für einen Mehrwert in der Darstellung, in eine Arc Diagram Darstellung zu integrieren.

Zoom

Zooming nennt man den Vorgang der zunehmenden Vergrößerung, bei gleichzeitiger Abnahme des darstellbaren Teilausschnittes einer Darstellung oder umgekehrt, unter der Bedingung, dass das Sichtfenster dabei eine konstante Größe behält [Sp06]. Der (graphische) Zoom ändert also die Größe der einzelnen Objekte in der Darstellung. Das mittlere Bild in Abbildung 2-3 versucht diese Möglichkeit zu verdeutlichen. Drei unterschiedliche Sichtfenster 1, 2 und 3 (unten) stellen einen Teil des Gesamtbildes (oben) vergrößert dar.

Eine andere Form des Zooms ändert die Detailstufe der Darstellung in Abhängigkeit vom Skalierungsgrad. Diese Technik wird als Semantischer Zoom bezeichnet. Das rechte Bild in Abbildung 2-3 zeigt, wie von der oberen zur unteren Darstellung der Detailgrad erhöht wird. Eine zusätzliche Beschriftung, eine Textur für die Ellipse und ein Farbverlauf für das ursprünglich nur gelbe Dreieck sind erkennbar.

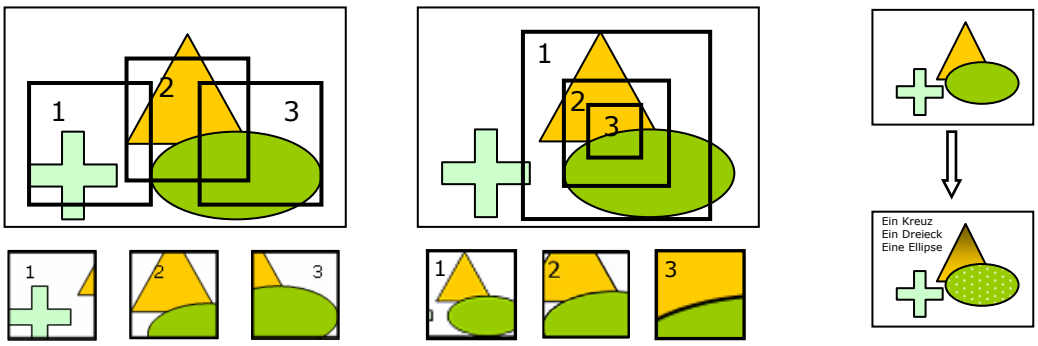

Abbildung 2-3: Veranschaulichung des Panning (links), Zooming (mittig) und des Semantischen Zooms (rechts)

Die Möglichkeit sich mit Hilfe des Zooming einen Überblick, als auch sehr feine Details anzeigen zu lassen, verdeutlicht rasch seinen Nutzen. Der Semantische Zoom ist dann besonders nützlich, wenn es gelingt, die zu visualisierenden Daten auf eine geeignete Weise zu gewichten. In Abhängigkeit dieses Entscheidungsmaßes können so in den einzelnen Skalierungsstufen schrittweise Zusatzinformationen oder feinere Details der entsprechenden Darstellung eingeblendet werden.

Einige dieser Maße werden in 4.4 im Rahmen des vorgestellten Konzepts diskutiert. Für die Steigerung der Qualität dieses Konzeptes kommt sowohl das Scrolling und Panning als auch das Graphische und Semantische Zooming in Betracht. Es wird zu untersuchen sein, auf welche Art und Weise die Konzepte den Nutzer bestmöglich unterstützen.

Visual Information Seeking Mantra

Durch die Vorstellung einiger grundlegender Konzepte der Visualisierung, wird an dieser Stelle bereits eines deutlich: Es sind mehr Informationen darzustellen, als dies auf einem Display möglich ist. Dies ist einer der Gründe warum die Visualisierung ein iterativer und interaktiver Prozess sein muss, in dem es gilt, eine Reihe geeigneter Darstellungen zu erzeugen. Es wird nicht möglich sein, mit einer Darstellung die Exploration der gesamten Datenmengen zu erreichen. Bertin [Be81] formulierte bereits 1981 treffend:

> *„A graphic is not drawn once and for all"*

Ein grundsätzliches Vorgehen, welches Shneiderman [Sh96] vorgeschlagen hat, versucht diesem Anliegen Folge zu leisten. Es lautet:

> *„Overview first, zoom and filter, then details on demand"*

Dieses „Mantra der Informationsvisualisierung" setzt die Möglichkeit der Erzeugung verschiedener Darstellungen voraus. Dies wiederum erfordert ein hohes Maß an Interaktionsfunktionalität. Mit dem Scrolling und Panning sowie dem Zooming wurden bereits zwei solcher Techniken kurz vorgestellt.

Nach Shneiderman ist das Visual Information Seeking Mantra unter anderem ein exzellenter Ausgangspunkt, um Werkzeuge und Benutzerschnittstellen für die visuelle Analyse von Daten zu entwerfen. Deshalb orientiert sich, das im Zuge dieser Arbeit entstandene interaktive Tool zur Visualisierung von multivariaten Daten durch Arc Diagrams, an diesem Ansatz. Auf welche Weise die einzelnen Stufen des Mantras dabei umgesetzt wurden, wird Gegenstand der Diskussion des fünften Kapitels sein.

2.2 Einflussfaktoren einer Visualisierung

Bisher wurde die Visualisierung als zielgerichtete Transformation von Daten in ein sichtbares Bild beschrieben, mit dem allgemeinen Ziel, den Erkenntnisprozess eines Betrachters zu beschleunigen. Dabei wurde der Prozess, als auch verschiedene Anforderungen an die Darstellung erläutert. Ob eine Visualisierung den genannten Anforderungen genügen kann, ist dabei von einer Reihe von Einflussfaktoren abhängig. Verschiedene Faktoren erfordern unterschiedliche Herangehensweisen an die Visualisierung, damit sie ihr kommunikatives Ziel erreicht. Zwei besonders wichtige Einflussfaktoren sollen vorgestellt werden: Die Charakteristika der Daten und das Bearbeitungsziel einer Visualisierung. Die Charakteristika einer Datenmenge sind der wichtigste Aspekt für den Visualisierungsprozess. Anhand der vorliegenden Daten werden die meisten Visualisierungsentscheidungen getroffen.

2.2.1 Die Beschreibung der Daten

Der Beobachtungsraum

Der Raum in dem die Daten erhoben werden, soll als *Beobachtungsraum* bezeichnet werden. An dieser Stelle erfolgt eine Abstraktion von der Art der Datenerhebung. Für die weiteren Untersuchungen ist es nicht von belang, ob die Daten tatsächlich beobachtet, gemessen oder entworfen sind. Auch die Tatsache, ob es sich tatsächlich um einen physikalischen Raum handelt aus dem die Daten stammen, ist nicht von Bedeutung.

Nach [SM00] sind vor allem drei Charakteristika des Beobachtungsraumes für eine Visualisierung interessant:

- Die Dimensionalität,
- der Wirkungskreis und
- der Verbund der Beobachtungspunkte.

Für das Ziel dieser Arbeit ist lediglich die Dimensionalität des Beobachtungsraumes von unmittelbarem Belang. Die Dimensionen, die den Beobachtungsraum aufspannen, sollen als *unabhängige Variable* bezeichnet werden.

Bei den vorliegenden Daten handelt es sich um Zeitreihen. Die einzig vorhandene unabhängige Variable ist die Zeit. Nach Chatfield [Ch82] ist eine Zeitreihe allgemein eine Sammlung von Daten, die in zeitlicher Abfolge beobachtet werden. Eine Zeitreihe wird *kontinuierlich* genannt, wenn die Beobachtungen kontinuierlich in der Zeit erfolgen, was bei einigen mechanischen Aufzeichnungsvorrichtungen der Fall ist. Als *diskret* wird eine Zeitreihe bezeichnet, wenn die Beobachtungen nur zu bestimmten *Zeitpunkten* vorgenommen werden. Unter einem Zeitpunkt seien hier Werte auf einer geordneten Zeitachse verstanden, welche keine Ausdehnung in der Dimension Zeit besitzen. [SM00] unterscheiden

zwischen dem Zeitbezug der Daten und dem in der Darstellung. Auf den Zeitbezug in der Darstellung wird am Anfang des dritten Kapitels eingegangen. Der Zeitbezug in den Daten wird in [SM00] genauer klassifiziert in:

- statischen,
- quasistatischen und
- dynamischen

Zeitbezug.

Ein statischer Zeitbezug der Daten ist demnach gegeben, wenn die Zeitachse nur einen festen Gültigkeitszeitpunkt für gegebene Werte hat. Quasistatisch sind Daten, für welche die Zeitachse mehrere diskrete Gültigkeitszeitpunkte aufweist. Dynamisch ist ein Zeitbezug der Daten genau dann, wenn die Zeitachse kontinuierlich skaliert ist.

In der vorliegenden Arbeit ist der Zeitbezug der Daten über die Speicherung eines Datums für jeden der Beobachtungsfälle hergestellt. Da mehrere Beobachtungsfälle auf einer tagesgenau diskretisierten Zeitachse vorliegen, handelt es sich also um quasistatische Daten. Es sei erwähnt, dass die Verwendung eines Datums als diskreter Zeitpunkt nicht für jedermann intuitiv verständlich ist. Im Allgemeinen wird z.B. eine sekundengenaue Uhrzeit als Zeitpunkt verstanden, nicht aber ein Tag. Tatsächlich bestimmt die zeitliche Auflösung der Zeitachse darüber, ob eine Zeitangabe als Zeitpunkt verstanden werden kann oder nicht.

Schlittgen und Streitberg [SS99] fassen die Beschreibung einer Zeitreihe in einer aussagekräftigen Definition zusammen:

> *„Eine (zeitlich) geordnete Folge $(F_t)_{t \in B}$ von Beobachtungen einer Größe wird als Zeitreihe bezeichnet. Für jeden Zeitpunkt t einer Menge B von Beobachtungszeitpunkten liegt dabei genau eine Messung vor.*"

In den vorliegenden Zeitreihen ist die Parametermenge *B* demnach eine endliche, diskrete Menge von äquidistanten Zeitpunkten. Es sei erwähnt, dass dies aber nicht immer so sein muss. Der interessierte Leser sei an dieser Stelle auf [Ch82] oder auch [SS99] für eine ausführlichere Diskussion verwiesen.

Die Merkmale

Die verschiedenen Parameter, die in einem Beobachtungsraum erfasst werden, bezeichnet man als *Merkmale*. Der konkrete Wert, den ein Merkmal in einem Beobachtungspunkt annimmt, wird *Ausprägung* genannt. Die Werteverteilung der Merkmale hängt von ihrer Verteilung im Beobachtungsraum ab. Merkmale werden deshalb auch als die *abhängigen Variablen* bezeichnet.

[SM00] nennen für Merkmale eine Reihe wichtiger Aspekte:

- Datentyp
- Dimensionalität
- Wertebereich
- Strukturierung

Der Datentyp soll als die Charakterisierung der Anzahl der Komponenten eines Merkmals verstanden werden: *Skalare Größen* sind durch einen Betrag vollständig charakterisiert. *Vektorielle Größen* repräsentieren neben einem Betrag auch noch eine Richtung. *Tensorielle Größen* sind eine Zusammenfassung skalarer Komponenten mit bestimmten Eigenschaften.

Die Anzahl der erfassten Merkmale in einem Beobachtungspunkt legt die Dimensionalität der abhängigen Variablen fest. Liegt für jeden Beobachtungspunkt genau ein Merkmal mit genau einer Ausprägung vor, so spricht man von *univariaten Daten*. Liegt mehr als ein einziges Merkmal vor, so wird von multivariaten Daten gesprochen.

Der Wertebereich von Merkmalen kann nach [SM00] in *qualitativ* und *quantitativ* unterschieden werden. Qualitative Merkmale verwenden, im Gegensatz zu quantitativen Merkmalen, nicht-metrische Skalen. Metrische Skalen werden in Intervall- und Verhältnisskala getrennt. Mit Hilfe von nominalen Skalen lässt sich lediglich eine Gleichheit oder Ungleichheit von Merkmalen feststellen. Ordinale Skalen erlauben es darüber hinaus, die Richtung verschiedener Ausprägungen aufgrund einer Ordnungsrelation zu ermitteln. Bei intervallskalierten Merkmalen besitzt zusätzlich die Differenz zwischen den Ausprägungen einen Informationsgehalt. Verhältnisskalen unterscheiden sich von Intervallskalen nach [Ba06] dadurch, dass für die möglichen Merkmalsausprägungen zusätzlich ein natürlicher Nullpunkt existiert.

Wichtig wird eine Klassifizierung des Wertebereichs unter anderem dann, wenn mathematische Operationen oder statistische Maße auf den Ausprägungen der Merkmale ausgeführt werden sollen. Im vierten Kapitel soll der Einsatz von multivariaten statistischen Analyseverfahren diskutiert werden. Unterschiedliche Verfahren setzen dabei verschiedene Skalierungsstufen voraus.

Ebenfalls von Interesse für diese Arbeit ist der *Umfang eines Wertebereichs*. Er beschreibt, welche möglichen Ausprägungen ein Merkmal annehmen kann. Für die Suche nach Wiederholungen in einer Zeitreihe und somit zum Finden von Mustern, hat dies einen direkten Einfluss. Je größer der Wertebereich, oder genauer: je größer das Verhältnis des Wertebereichsumfangs zur Länge der Zeitreihe, desto kleiner ist die Chance, sich wiederholende Werte in einer Zeitreihe zu finden. Auf welche Weise diese Chance gesteigert werden kann und wann dies sinnvoll ist, wird im vierten Kapitel diskutiert.

Für die Strukturierung von Merkmalen sind nach [SM00] die Varianten

- Sequentiell, Speicherung in einer Liste
- Relational, Speicherung in Tabellenform
- Hierarchisch, Speicherung in Baumstruktur
- Netzwerkartig, Speicherung in Netzwerkform

zu unterscheiden.

Anhand der eingeführten Aspekte wird die zugrunde liegende Zeitreihe dieser Arbeit nun vorgestellt.

Tabelle 2-1: Die multivariate Zeitreihe

Beobachtungs-Raum **Unabhängige Variable**	**Merkmale** **Abhängige Variablen**			
Tag	Höchsttemp. [°C]	rel. Luftf. [%]	…	Sonnenschein [h]
01.01.1893	*-8.1*	*79*	…	*2.6*
02.01.1893	*-8.5*	*97*	…	*1.1*
03.01.1893	*-9.7*	*85*	…	*0.2*
04.01.1893	*-4.7*	*96*	…	*0*
…	…	…	…	…
31.08.2003	*18.5*	*80*	…	*10*

In Tabelle 2-1 ist ein geringer Teil der vorliegenden multivariaten Zeitreihe beispielhaft abgebildet. Für die Zeitreihe liegen in den Beobachtungspunkten des Beobachtungsraumes jeweils zehn Merkmale mit jeweils genau einer Ausprägung pro Beobachtungspunkt vor. Tabelle 2-1 zeigt drei dieser Merkmale mit einigen Ausprägungen. Weil mehr als ein Merkmal vorliegt, wird im Folgenden von einer *multivariaten Zeitreihe* gesprochen.

Die multivariate Zeitreihe erfasst alle zehn Merkmale im Zeitraum vom 01.01.1893 bis zum 31.08.2003. Basierend auf einer tagesgenau diskretisierten Zeitachse liegen somit für jedes Merkmal genau 40419 Ausprägungen vor. Die Umfänge der Wertebereiche der einzelnen Merkmale unterscheiden sich dabei teilweise erheblich.

Alle auftretenden Merkmale sind skalaren Datentyps und können als metrisch klassifiziert werden. Sie sind also mindestens intervallskaliert. Zum Beispiel ist das Merkmal „Höchsttemperatur" eines Tages aus Tabelle 2-1 ein intervallskaliertes Merkmal.

Die multivariate Zeitreihe ist relational gespeichert. Eine solche Form ist deshalb auch in Tabelle 2-1 skizziert.

2.2.2 Bearbeitungsziele

Neben den behandelten Charakteristika der Daten sind weitere Faktoren für den Erfolg einer Darstellung zu berücksichtigen. Ein weiterer wichtiger Aspekt ist das Bearbeitungsziel. Es beschreibt die konkrete Problemstellung, welche durch eine Visualisierung gelöst werden soll. Beshers und Feiner [BF92] nennen drei primäre Visualisierungsziele:

- Exploration,
- Vergleich und
- Direkte Suche.

Diese drei Stufen sind angelehnt an die Fähigkeiten des menschlichen visuellen Systems. Es ist dem Mensch möglich, seine Aufmerksamkeit auf das gesamte Bild zu richten, Gruppen von Bildern zu vergleichen oder seine Aufmerksamkeit auf einzelne Teile zu fokussieren. Die Exploration ist speziell darauf angelegt Zusammenhänge in den Daten zu entdecken. Dabei wird davon ausgegangen, dass ein Nutzer a priori keine oder nur sehr wenige Kenntnisse über die in den Daten enthaltenen Zusammenhänge hat. Beim Vergleich geht es mehr darum vermutete Sachverhalte zwischen Daten durch eine visuelle Gegenüberstellung von, auf den Daten beruhenden, Bildern zu untermauern oder zu widerlegen. Die direkte Suche erfordert a priori eine relativ genaue Vorstellung der Daten. Ähnlich wie beim Vergleich sollen primär Strukturen geprüft werden. Nun allerdings nicht vergleichend, sondern auf einzelne Teile beschränkt.

In der vorliegenden Arbeit soll mit Hilfe der Arc Diagram Technik eine visuelle Exploration gegebener Datenmengen erfolgen. Diese gegebene Datenmenge wurde im vorherigen Abschnitt als multivariate Zeitreihe identifiziert. Eine Möglichkeit zur Exploration dieser Zeitreihe ist es Muster zu identifizieren und darzustellen. Genau dafür wird im Rahmen dieser Arbeit eine effektive Vorgehensweise entwickelt. Unter dem Begriff „Muster" sei hier zunächst intuitiv eine sich wiederholende Folge von Ausprägungen eines Merkmals innerhalb einer Zeitreihe verstanden. Eine exaktere Definition wird an passender Stelle im vierten Kapitel erfolgen.

Zusammenfassung und Fazit

In diesem Kapitel wurde die grundsätzliche Zielsetzung einer jeden Visualisierung beschrieben. Eine geeignete Visualisierung stellt dabei genau die in den Daten enthaltenen Aspekte dar und schließt auf diese Weise eine eventuelle Fehlinterpretation der Daten aus. Zur Erzeugung einer solchen Darstellung wurde ein dreistufiger Prozess skizziert, welcher als Visualisierungspipeline bezeichnet wurde. Es wurde erläutert, dass in der Regel mehr Daten darzustellen sind, als auf einem Anzeigegerät Platz zur Verfügung steht. Um dieses Problem zu reduzieren, wurden einige Basiskonzepte der Visualisierung vorgestellt. Aufbauend auf der dargestellten Zielsetzung sollen drei Verfahren zur

Visualisierung von Mustern vorgestellt werden. Die erfolgte Beschreibung der Daten liefert die Grundlage für die spezielle Untersuchung der Arc Diagrams zur Mustersuche in multivariaten Zeitreihen.

Kapitel 3

Verfahren zur Visualisierung von Mustern

Dieses Kapitel hat die Aufgabe, Möglichkeiten und Probleme bestehender Mustervisualisierungsverfahren aufzuzeigen. Verfahren zur Darstellung von Mustern werden in vielen Anwendungsbereichen eingesetzt. Demzufolge existiert eine große Zahl solcher Verfahren. Exemplarisch werden drei Verfahren vorgestellt. Die H-Curves [HR83] zur Darstellung von Mustern im Bereich der Genanalyse, die Dotplots [CH92] u.a. zur Musterdarstellung in Programmierungscode und die Arc Diagrams [Wa02] allgemein zur Musterdarstellung in Zeichenfolgen aber auch speziell in Musikstücken.

Diese drei Techniken werden stellvertretend ausgewählt, da sie aus unterschiedlichen Anwendungsbereichen stammen und im Hinblick auf ihren Zeitbezug in der Darstellung statische Techniken sind. Sie ändern sich nicht automatisch über die Zeit. Im Gegensatz zu dynamischen Techniken lassen sie insbesondere quantitative Aussagen zu. Dies ist im Umfeld der Darstellung von Mustern eine wichtige Eigenschaft, da qualitative Aussagen in der Regel nicht ausreichen. Darüber hinaus sind statische Techniken nach [SM00] für Daten mit quasistatischem Zeitbezug, und somit für vorliegende Zeitreihe, gut geeignet.

Am Ende des Kapitels werden die Verfahren im Hinblick auf ihre unterschiedlichen Leistungsfähigkeiten und Probleme verglichen. Der Aufgabenstellung dieser Arbeit folgend, soll der besondere Fokus auf den zu erwartenden speziellen Problemen der Arc Diagram Technik beim Übergang der Visualisierung von Zeichenfolgen zu multivariaten Zeitreihen liegen.

3.1 H-Curves

Die H-Curves, eines der ersten Verfahren, welches für die visuelle Analyse von DNA-Sequenzen vorgesehen ist, stellten Hamori und Ruskin bereits 1983 vor. Die Idee dieser Visualisierungsmethode besteht darin, den Informationsgehalt von Folgen von Nukleinsäuren aus ihrer textlichen Beschreibung A (Adenin), C (Cytosin), G (Guanin) und T (Thymin) als die vier möglichen Basen auf eine 3-dimensionale

Raumkurve, genannt H-Curve, abzubilden. DNA-Moleküle liegen als eine Doppelhelix mit zwei komplementären Strängen vor. Durch die Basenpaare A–T und G–C sind sie aber eindeutig festgelegt. Es genügt daher zur vollständigen Beschreibung einer Sequenz genau einen Komplementärstrang anzugeben [NM99]. Dazu geht diese Visualisierungstechnik wie folgt vor:

Die positive Z-Achse des 3-dimensionalen Kartesischen Koordinatensystems wird zum Zählen der einzelnen Nukleotide, der kleinsten Bausteine der Nukleinsäuren, verwendet. Zur Darstellung wird die Positionsangabe auf der vertikal verlaufenden Z-Achse abgetragen und es wird pro Base ein Vektor (1, 1, 0) für A, (-1, 1, 0) für G, (-1, -1, 0) für C und (1, -1, 0) für T zu (0, 0, n) addiert. Der Index *n* wird zum Zählen schrittweise von der maximalen Anzahl betrachteter Nukleotide bis null verringert. Abbildung 3-1 zeigt rechts das zugrunde liegende 3-dimensionale Kartesische Koordinatensystem. Auf der linken Seite sind die möglichen Vektoren der vier Basen mit ihrer entsprechenden Orientierung eingetragen.

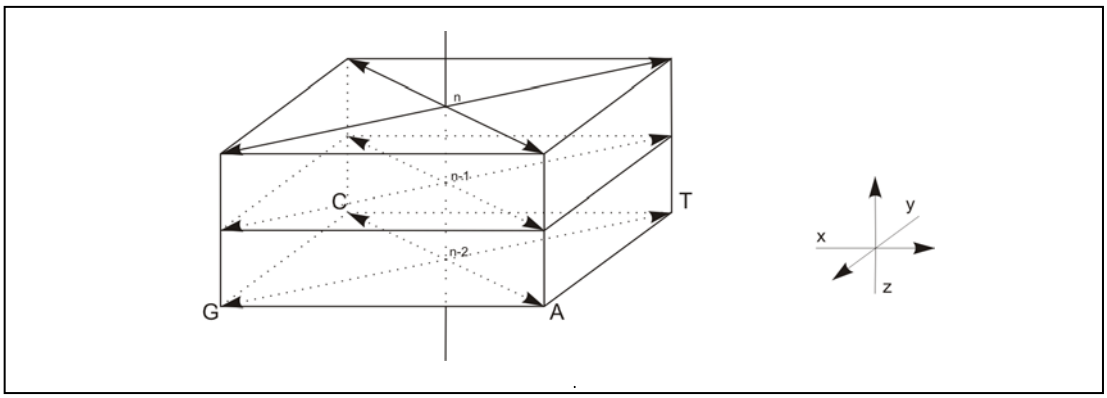

Abbildung 3-1: Die vier Basisvektoren bei der 3-dimensionalen H-Curve Darstellung, erstellt mit CorelDRAW

Für die Erstellung der gesamten Darstellung heißt das, dass für jeden Punkt der Z-Achse in der korrespondierenden XY-Ebene ein Schritt in die entsprechende Richtung A, G, C oder T erfolgt und dort ein Punkt abgetragen wird. Die abgetragenen Punkte werden dann durch einen Linienzug miteinander verbunden. Die Wiederholung, also ein wiederkehrendes Muster, einer Folge von Nukleotiden wird sichtbar durch entstehende geometrische Figuren. AGCT in mehrfacher Wiederholung werden beispielsweise eine Spirale um die Z-Achse erzeugen, andere Muster erzeugen entsprechend andere Figuren. Abbildung 3-2 zeigt eine mögliche H-Curve.

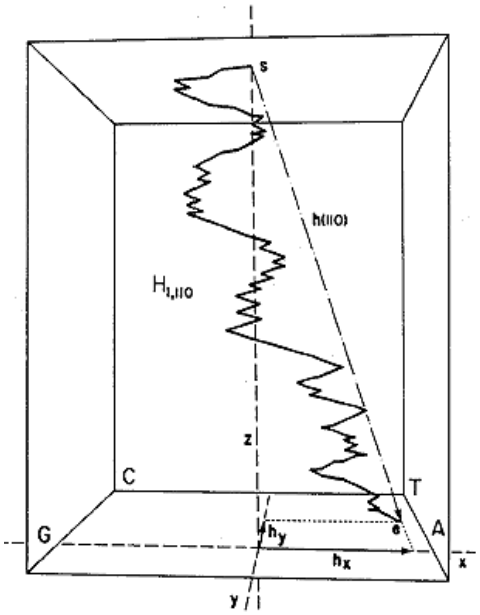

Abbildung 3-2: H-Curve, aus [HR83]

Ein Vorteil der H-Curves ist, dass man sehr schnell erkennt, welche Symbole in der Zeichenfolge überwiegen. Möglich wird dies durch die Darstellung der „Hauptrichtung" des Linienzuges. In Abbildung 3-2 tendiert zum Beispiel die Kurve in Richtung des Thymins. Nachteilig wirkt sich in dieser Darstellung aus, dass nur schwer wiederkehrende Zeichenfolgen identifiziert werden können. Ein wichtiger Grund dafür ist, dass die Ausgabe der 3-dimensionalen Figuren in der Regel auf einem 2-dimensionalen Gerät erfolgt. Bei der Darstellung von univariaten Zeitreihen würde die Z-Achse zur Darstellung des Zeitbezugs verwendet werden. Der Zeitverlauf ist dabei entgegen der Z-Achse (in Abbildung 3-2 von oben nach unten) festgelegt. Wenn die Zeitreihen diskreten Charakter haben, so sollten die diskreten Zeitpunkte auf der Z-Achse auch nicht durch einen Linienzug verbunden werden. Dies würde die Expressivität der Darstellung beeinträchtigen. Ohne einen verbindenden Linienzug fällt es jedoch noch schwerer, Muster in der Darstellung zu erkennen.

3.2 Dotplots

Dotplot ist ein interaktives Computerprogramm, welches von Church und Helfman 1992 vorgestellt wurde. Es dient dem Zweck, Textzeilen und Programmcode nach Mustern zu durchsuchen. Der Ansatz zur Darstellung solcher Muster ist nach [CH92] der Biologie entliehen. Dort geht es darum Selbstähnlichkeiten in DNA Sequenzen aufzuspüren. Church und Helfman wollten ein Werkzeug bereitstellen, welches dabei hilft, Strukturen vor allem in extrem großen Strings zu identifizieren.

Im Prinzip ist ein Dotplot eine visuelle (Autokorrelations-) Matrix. Abbildung 3-3 zeigt, wie die zu betrachtende Zeichenfolge in Form einer Matrix dargestellt wird: Der allgemeinen Schreibweise für lateinische Alphabete folgend, wird der String entsprechend einmal von links nach rechts und von oben nach unten abgetragen. Der grundsätzliche Ablauf des Algorithmus ist Folgender:

1. Zerlegen einer Zeichenfolge in Teilfolgen (z.B. Zerlegung in einzelne Wörter wie in Abbildung 3-3 links)
2. Platzierung eines Punktes an der Position *(i,j),* wenn die *i`te* Teilfolge mit der *j`ten* Teilfolge übereinstimmt.

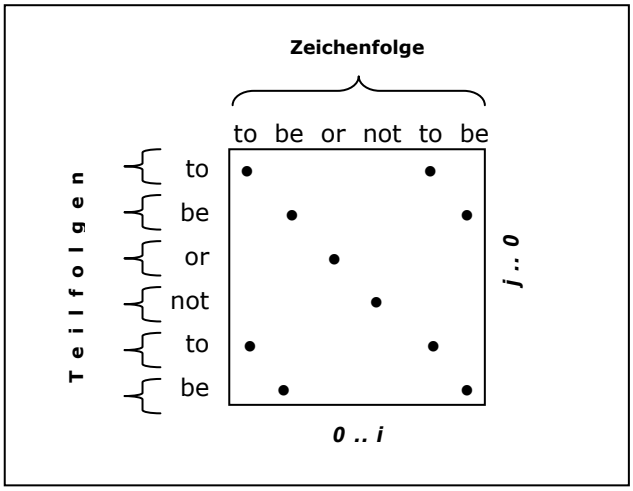

Abbildung 3-3: Eine Dotplot Darstellung, sechs Wörter von Shakespeare, angelehnt an [He96]

Die Identifizierung der Muster in der Zeichenfolge wird auch hier durch das Entstehen von unterschiedlichen geometrischen Formen ermöglicht, welche beim Zeichnen der Punkte entstehen. Diagonalen, Quadrate und Texturen sind solche Hinweise auf Ähnlichkeiten innerhalb einer Zeichenfolge. Abbildung 3-4 zeigt auf der rechten Seite wie Diagonalen beispielsweise auf Regionen mit geordneter Ähnlichkeit hinweisen. Die linke Seite der Abbildung 3-4 zeigt Quadrate, welche ein Indiz für eine ungeordnete Ähnlichkeit sind.

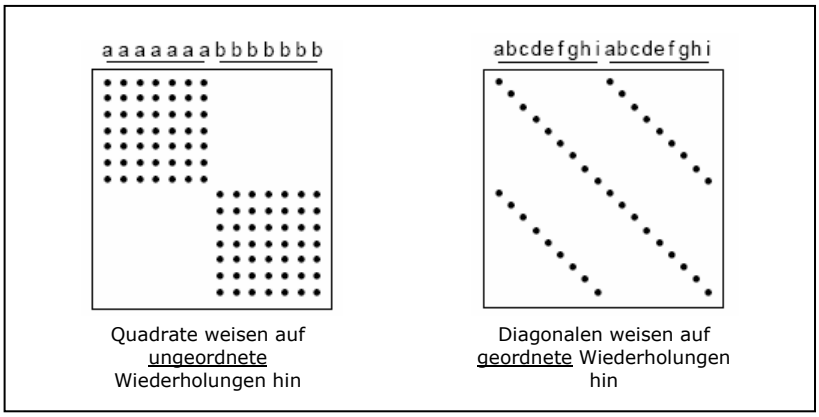

Abbildung 3-4: ungeordnete und geordnete Wiederholungen von Mustern durch Dotplots, aus [CH92]

Darüber hinaus wurde die Dotplot Technik von Church und Helfman um weitere Fähigkeiten und Verbesserungen ergänzt. Das sich wiederholende Muster z.B. wurde, wie in Abbildung 3-4 ersichtlich, zusätzlich unterstrichen und es wurden eine Reihe von Möglichkeiten vorgestellt, das Verfahren zu beschleunigen. Diese Diskussion soll hier nicht weiter vertieft werden. Es sei auf [CH92] verwiesen, um weitere Einzelheiten zu erfahren.

Günstig an der Dotplot Technik ist, dass interessante Bereiche in vergleichsweise großen Zeichenfolgen recht gut erkannt werden können. Probleme hat die Dotplot Technik dann, wenn häufige Wiederholungen derselben Muster auftreten. Abbildung 3-4 beispielsweise zeigt auf der linken Seite, dass durch eine 7-malige Wiederholung des Zeichens „a" 49 Punkte in der Matrix darzustellen sind. Die Matrixanordnung bedingt diese quadratische Abhängigkeit. Treten häufig wiederholte Zeichenfolgen auf, kann die Dotplot Darstellung dadurch schnell unübersichtlich werden. Sollen die Dotplots zur Darstellung von Zeitreihen eingesetzt werden, so ist der zeitliche Verlauf entlang der Hauptdiagonalen in der Matrix (von oben links nach unten rechts) ablesbar. Der horizontale Abstand einer Ausprägung zur Hauptdiagonalen kann als Zeitspanne zwischen dem Auftreten eines Musters interpretiert werden. Schließlich sei noch erwähnt, dass das Setzen der Punkte in der Matrix einen diskreten Charakter der dargestellten Reihe suggeriert. Für kontinuierliche Reihen sollten Dotplots daher keine Verwendung finden.

3.3 Arc Diagrams

Die Arc Diagram Technik von Wattenberg [Wa02] stellt einen wesentlichen Schwerpunkt bei der Vorstellung bestehender Verfahren zur Mustervisualisierung dar und bildet auch die Grundlage dieser Arbeit.

Ebenso wie die zuvor beschriebenen Verfahren, haben auch die Arc Diagrams das Ziel, die Struktur einer Zeichenfolge zu zeigen, indem sie sich wiederholende Teilfolgen innerhalb einer Zeichenfolge hervorheben.

Die grundsätzlichen Regeln zur Konstruktion eines Arc Diagrams einer Zeichenfolge *S* mit der Länge *k* lauten wie folgt: Zunächst wird *S* auf die X-Achse abgebildet: Das Symbol mit dem Index *l* der Zeichenfolge entspricht dabei der Position *(l/k, 0)* im 2-dimensionalen Kartesischen Koordinatensystem. Durch diese Abbildung entstehen auf der X-Achse Intervalle, die mit den Paaren von Teilzeichenfolgen *Z* aus *S* korrespondieren. Die korrespondierenden Paare von Mustern, *X* und *Y*, werden durch einen breiten halbkreisförmigen Bogen verbunden. Der innere Halbkreis verbindet dabei das Ende des Intervalls von *X* mit dem Anfang des Intervalls von *Y*. Der äußere Halbkreis verbindet den Anfang des Intervalls von *X* mit dem Ende des Intervalls von *Y*. Auf diese Weise entstehen Bögen, welche in der Höhe zum Abstand der korrespondierenden Paare von Mustern, *X* und *Y*, proportional sind.

In der Praxis werden Zeichenfolgen eine Vielzahl von sich wiederholenden Teilzeichenfolgen mit unterschiedlichen Längen enthalten. Es wird gezwungenermaßen zu Überlappungen und Überschneidungen von Bögen kommen. Daher schlägt Wattenberg vor, die Bögen mit Transparenz zu versehen, sodass kein Bogen komplett durch einen anderen verdeckt werden kann. Abbildung 3-5 zeigt 48 Paare von Mustern verbunden durch semitransparente Bögen.

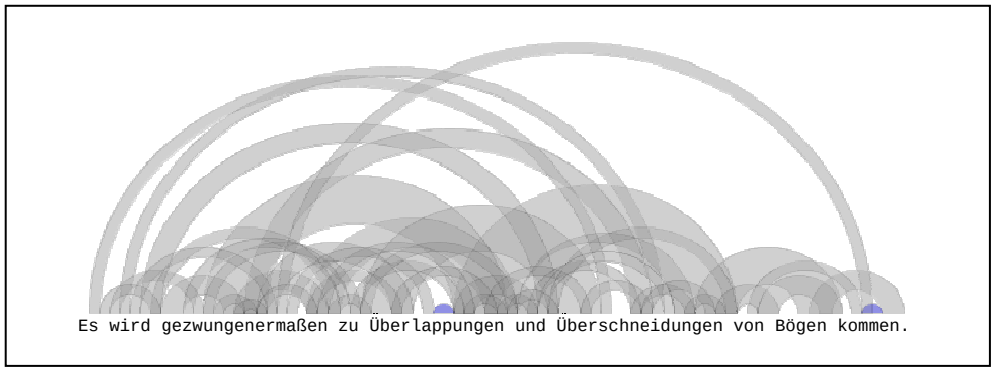

Abbildung 3-5: Arc Diagram Darstellung einer Zeichenfolge, erstellt mit eigener Implementierung

Um die Darstellung nicht zu überladen, ist in den Arc Diagrams eine neue Idee umgesetzt. Es ist nur eine Teilmenge aller möglichen Muster dargestellt. Die Auswahl dieser Teilmengen ist von entscheidender Bedeutung für die Erkennung der tatsächlichen Struktur einer Zeichenfolge. Um dem Nutzer das Sehen, Erkennen und Bewerten durch die Darstellung der „entscheidenden" Teilmenge von Musterpaaren zu ermöglichen, ist eine Reihe von Definitionen nötig [Wa02]:

> *Ein maximal passendes Paar* ist ein Paar von Teilfolgen *X* und *Y* einer Zeichenfolge *S,* welche:

1. *Identisch* sind. X und Y enthalten dieselbe Folge von Symbolen.
2. *Nicht-überlappend* sind. X und Y überschneiden sich nicht.
3. *Fortlaufend* sind. X steht vor Y und es gibt keine Teilfolge Z, identisch zu X und Y, deren Beginn zwischen den Beginn von X und den Beginn von Y fällt.
4. *Maximal* sind. Es gibt keine längeren identischen und nicht überlappenden Teilfolgen *X'* und *Y'*, mit $X \subseteq X' und Y \subseteq Y'$.

Des Weiteren ist die folgende Definition erforderlich:

> *Ein Wiederholungsbereich R* ist eine Teilfolge *R* von *S*, mit der Eigenschaft, dass *R* aus einer sich mindestens zweimal fortlaufend wiederholenden Zeichenfolge *P* entsteht. Jede Wiederholung von *P* wird dabei als *fundamentale Teilfolge* von *R* bezeichnet.

Die letzte Definition legt fest, welche Paare von Teilfolgen der Zeichenfolge *S* tatsächlich als Arc Diagram dargestellt werden. Solche Paare sollen als *essentiell passendes Paar* bezeichnet werden.

> Ein essentiell passendes Paar ist ein Paar von Teilfolgen *X* und *Y* von *S*, welche:
>
> 1. Ein maximal passendes Paar sind, welches nicht in einem Wiederholungsbereich enthalten ist,
> 2. *Oder*, ein maximal passendes Paar sind, enthalten in der selben fundamentalen Teilfolge eines beliebigen Wiederholungsbereiches, die es enthält,
> 3. *Oder,* zwei fortlaufende fundamentale Teilfolgen eines Wiederholungsbereiches.

In Abbildung 3-5 sind essentiell passende Paare dargestellt. Bei den grauen Bögen handelt es sich um maximal passende Paare (Typ 1). Die beiden kleineren blauen Bögen weisen auf einen Wiederholungsbereich hin (Typ 3).

Trotz der Transparenz und der Idee, nur eine Teilmenge aller möglichen Muster darzustellen, bleibt ein Problem bestehen: Zeichenfolgen der Länge *k*, die aus einem zu *k* relativ kleinen Alphabet *A* aufgebaut sind, werden sehr viele kurze, sich wiederholende Teilfolgen enthalten. Wattenberg empfiehlt eine untere Schranke für die Länge der darzustellenden Teilfolgen einzuführen. Dies bewirkt, dass kurze Folgen nicht die Identifizierung, wahrscheinlich wichtigerer, längerer Musterreihen erschweren. In Abbildung 3-6 sind auf der linken Seite alle

essentiell passenden Paare dargestellt. Die rechte Seite zeigt nur zwei Paare. Hier ist die untere Schranke zur Darstellung auf größer gleich fünf zusammenhängende Zeichen festgesetzt. Es wird deutlich, dass die beiden Muster der rechten Darstellung auf der linken Seite vergleichsweise schwer zu identifizieren sind. Es kommt zu Überdeckungen und Überschneidungen von Bögen. Dies stellt einen Nachteil der Arc Diagram Technik dar. Auch ist der Aufwand zur Berechnung eines essentiell passenden Paares vergleichsweise hoch. So müssen zunächst maximal passende Paare und Wiederholungsbereiche bestimmt werden. Danach kann entschieden werden, ob es sich auch um essentiell passende Paare handelt. Ein wichtiger Vorteil dadurch ist, dass Darstellungen nicht so schnell überladen werden. Insbesondere für Zeitreihen ist interessant, dass die Höhe eines Bogen direkt proportional zum Abstand der Musterpaare ist. Je länger die Zeitspanne zwischen einem essentiell passenden Paar ist, desto höher ist der dazu korrespondierende Bogen in der Darstellung.

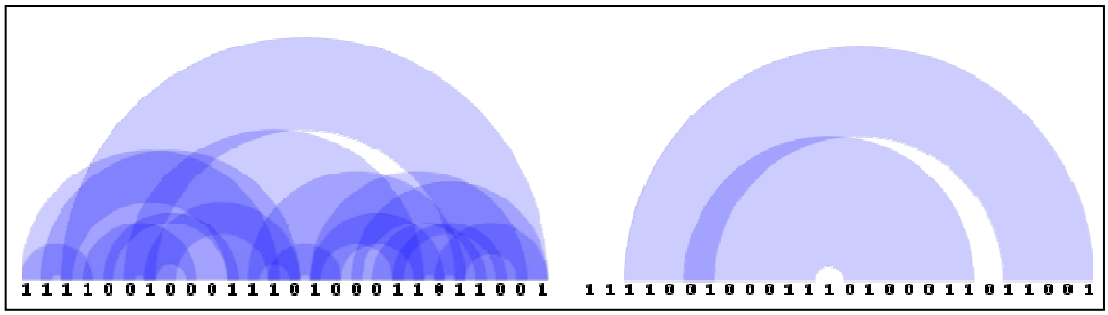

Abbildung 3-6: Zwei Arc Diagrams einer Zeichenfolge *S* mit der Länge *N*=26 und der Alphabetgröße *A*=2, erstellt mit eigener Implementierung

3.4 Vergleich der Techniken

Dieser Abschnitt soll zeigen, dass die Arc Diagram Technik für die Visualisierung von Mustern gut geeignet ist und daher berechtigt als Grundlage dieser Arbeit dient. Dazu werden die bislang vorgestellten Verfahren vergleichend gegenübergestellt.

Dotplots und Arc Diagrams sind 2-dimensionale Techniken. Dies stellt einen Unterschied zu H-Curves dar, bei der es sich um eine 3-dimensionale Technik handelt. 2-dimensionale Techniken lassen es im Allgemeinen zu, konkrete Werte exakt abzulesen. Dies ist bei 3-dimensionalen Darstellungstechniken weniger gut möglich.

Die Möglichkeit der Identifizierung der einzelnen Muster ist damit eng verbunden. Die H-Curve als 3-dimensionale Technik ist zwar in der Lage, Details zu zeigen, aber diese sind durch einen Betrachter nur schwer zu interpretieren. Auch wenn markante Wendungen in den H-Curves ins Auge fallen, lassen sich kurze wiederholte Muster weniger gut ausmachen. Die Identifizierung von Mustern ist in Arc Diagrams hingegen

gut möglich, weil z.B. die Darstellung durch Wiederholungen von Mustern nicht so schnell überladen wird.

Bei allen drei vorgestellten Verfahren handelt es sich um vollständige Darstellungen. Es werden also alle gegebenen Datenmengen in einem Bild dargestellt. Dies bietet den großen Vorteil, dass man auf einen Blick alle relevanten Informationen sehen kann. Das Verstehen und Bewerten ist dagegen häufig weitaus schwieriger, da für große Datenmengen das Bild schnell überladen wird. Die Darstellung ist dann nicht mehr interpretierbar. Um dies dennoch zu erreichen, gehen die Arc Diagrams einen Kompromiss ein. Anstatt, wie bei unvollständigen Darstellungen nur eine echte Teilmenge der Daten in einer Überblicksdarstellung zu repräsentieren und zunächst von feinen Details zu abstrahieren (vgl. Abschnitt 2.1.3), stellt die Arc Diagram Technik nur eine echte Teilmenge der gefundenen Muster dar. Genau solche Muster werden dargestellt, welche für das wesentliche Verständnis und die damit verbundene Exploration der Datenmenge von Bedeutung sind. Dies ist ein Unterschied zu den „echten" vollständigen Darstellungen der H-Curves und Dotplots. Diese bringen alle enthaltenen Muster zur Anzeige. Zugleich ist das aber auch der Grund dafür, dass die Arc Diagram Technik vergleichsweise besser skaliert, als etwa die Dotplot Technik. Speziell für häufig wiederholte Muster innerhalb einer Zeichenfolge ist die Arc Diagram Technik im Vorteil. Dafür sorgt die in Abschnitt 3.3 eingeführte Eigenschaft „Fortlaufend": Für w Wiederholungen von Teilfolgen X enthält ein Arc Diagram genau w-1 Bögen. Bereits an früherer Stelle wurde erwähnt, dass eine Dotplot Darstellung in derselben Situation eine quadratische Abhängigkeit aufweist.

Arc Diagrams bieten einen interessanten neuen Ansatzpunkt auf welche Art und Weise ein Muster visuell hervorgehoben wird. Die H-Curves und die Dotplots erlauben eine visuelle Erkennung eines Musters durch die Abbildung auf eine sich wiederholende geometrische Figur. Die Form der geometrischen Figur ist dabei direkt und allein abhängig von der Charakteristik des Musters (vgl. z.B. Abbildung 3-4). Der Ansatz der Arc Diagram Technik geht darüber hinaus. Zwar ist die Höhe und Breite eines Bogens direkt abhängig von der Entfernung und Länge eines Musters, aber der Bogen stellt zusätzlich einen direkten räumlichen Bezug zwischen den einzelnen wiederholten Teilfolgen dar. Er fungiert als eine Art Wegweiser zwischen gleichen wiederkehrenden Mustern. Dies ist von großem Vorteil, da ein Betrachter zwischen solchen Mustern rasch visuell eine Verbindung herstellen kann. Diese Tatsache untermauert Abbildung 3-7. Es wird auf den ersten Blick deutlich, dass es sich um Wiederholungen des gleichen Musters handelt. Durch die wegweisenden Bögen kann man sehr rasch eine optische Verbindung herstellen.

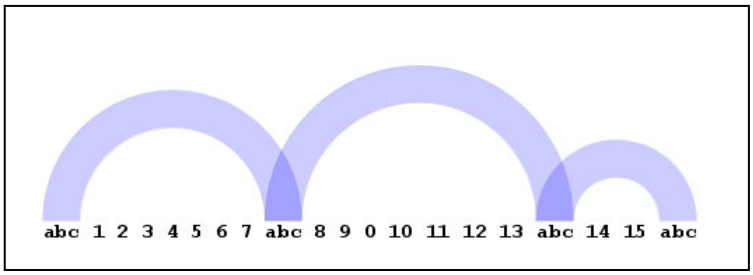

Abbildung 3-7: (w-1) Bögen eines (w-mal) wiederholten Musters "abc". Die Bögen fungieren dabei als Wegweiser zu wiederkehrenden Mustern.

Der Vergleich zeigt, dass alle Verfahren unterschiedliche Vor- und Nachteile haben. Um das Potential der Arc Diagrams auszuschöpfen, muss man die speziellen Möglichkeiten und Probleme der Technik kennen. Die wichtigsten Möglichkeiten und Probleme wurden im Laufe des Abschnitts 3.3 und dieses Abschnitts herausgearbeitet. Demnach sind Arc Diagrams zur Visualisierung von Mustern insbesondere durch ihre wegweisende Bogenform und ihre gute Skalierungsfähigkeit für häufig wiederkehrende Muster gut geeignet.

Die größte Herausforderung dieser Arbeit ist, dass Arc Diagrams für die Darstellung von Zeichenfolgen konzipiert sind. Die Aufgabenstellung dieser Arbeit ist allerdings, multivariate Zeitreihen mit Hilfe dieser Technik darzustellen. Sowohl für die Suche nach Mustern als auch für die Darstellung dieser, müssen deshalb neue innovative Ideen entwickelt werden.

Im weiteren Verlauf der Arbeit wird zu untersuchen sein, ob für ein Konzept zur Darstellung von multivariaten Zeitreihen mit Arc Diagrams, die Anzeige der gesamten Datenmenge in einem Bild, und somit dass Kriterium der Vollständigkeit, aufrechterhalten werden kann. Es ist zu klären, ob die neue Idee der Arc Diagrams, eine echte Teilmenge der Muster darzustellen, ausreicht, um die Interpretierbarkeit der Zeitreihen zu gewährleisten.

Weiterhin soll abgewogen werden, ob die Möglichkeiten der Verschlüsselung zusätzlicher Parameter durch eine dritte Dimension, die damit verbundenen Nachteile, z.B. der genannten, vergleichsweise schlechteren Identifizierung von Mustern in Arc Diagrams, aufwiegen kann.

Eine weitere Herausforderung ist die geeignete Integration von Interaktionstechniken. Dies ist in der Arc Diagram Technik ursprünglich nicht vorgesehen. Um nach Abschnitt 2.1.3 eine vollständige Exploration einer Datenmenge zu erreichen, ist die Integration von Interaktionstechniken aber unerlässlich.

Kapitel 4

Ein neues Konzept zur Mustervisualisierung multivariater Zeitreihen mit Hilfe der Arc Diagrams

Im letzten Kapitel wurden spezielle Techniken zur Visualisierung von Mustern in Zeichenfolgen vorgestellt. Es soll nun, am Beispiel der Arc Diagrams, ein Konzept vorgestellt werden, um diese Technik für die Visualisierung von Mustern in multivariaten Zeitreihen nutzbar zu machen.

4.1 Anforderungen

Zunächst muss dazu die Frage nach den grundlegenden Anforderungen an ein solches Konzept geklärt werden. Beim Übergang der Visualisierung von univariaten zu multivariaten Zeitreihen kommt es, ganz allgemein betrachtet, zu einer Reihe von neuen Herausforderungen. Die wichtigste Herausforderung dabei ist die Behandlung der wesentlich erhöhten Datenmenge. Diese Tatsache wirft im Umfeld der Visualisierung vor allem zwei Probleme auf:

- Die Verarbeitung der erhöhten Datenmenge muss in akzeptabler Zeit erfolgen.
- Die erhöhte Datenmenge ist in geeigneter Weise auf einem Ausgabegerät darzustellen.

Aufbauend auf diesen Forderungen, welche allgemeingültigen Charakter haben, und der Erschließung der bestehenden speziellen Probleme und Möglichkeiten der Arc Diagram Technik in den Abschnitten 3.3 und 3.4 werden hier drei fundamentale Lösungsansätze zur Visualisierung von multivariaten Zeitreihen auf Basis der Arc Diagrams vorgestellt:

1. Es muss eine Verbesserung und Erweiterung der Möglichkeiten der Mustersuche erfolgen. Dazu ist die bestehende Suche zu verbessern. Zusätzlich ist die Arc

Diagram Technik, um eine Möglichkeit zur Suche nach merkmalübergreifenden Zusammenhängen zu erweitern.

2. Die Darstellung von Mustern einer Zeitreihe muss effektiv auf dem begrenzten Platz eines Ausgabegerätes erfolgen. Dabei sind zwei Minimalanforderungen an die Darstellung gestellt:
   - Es soll zunächst mindestens ein Überblick über gefundene Muster gegeben werden.
   - Muster bezüglich eines Merkmals und Muster bezüglich mehrerer Merkmale müssen voneinander unterscheidbar sein.
3. Es muss gelingen, Interaktionstechniken in den Prozess der Mustersuche als auch der Mustervisualisierung zu integrieren. Dies muss so erfolgen, dass es einem Nutzer möglich ist, zielführend in beide Prozesse einzugreifen. Ein Grund dafür ist in den unterschiedlichsten Charakteristika zu verarbeitender Zeitreihen zu finden, für die automatisch gewählte Voreinstellungen nicht immer gut geeignet sind. Ein weiterer wichtiger Punkt ist, dass für die vollständige Exploration einer Menge von Daten eine einzelne Darstellung niemals ausreichend sein kann (vgl. Abschnitt 2.1.3).

Gelingt es akzeptable Lösungen für die drei genannten Punkte zu finden und diese Lösungen geeignet zu kombinieren, sodass sie als Einheit zusammenarbeiten, dann ist eine Möglichkeit gefunden, „Arc Diagrams" auch zur Visualisierung von multivariaten Zeitreihen einzusetzen. Dazu sollen nun die genannten Lösungsansätze, zunächst getrennt voneinander, betrachtet werden. Als Proof-of-Concept sollen die Lösungen im fünften Kapitel geeignet kombiniert werden.

4.2 Verbesserung und Erweiterung der Mustersuche

Die Arc Diagram Technik läuft im Prinzip in zwei Schritten ab. Der erste Schritt sucht alle essentiell passenden Paare. Der zweite Schritt stellt diese Paare dar. Der Abschnitt 4.2 widmet sich der Verbesserung und Erweiterung des Suchens von essentiell passenden Paaren. Zu diesem Zweck wird in Abschnitt 4.2.1 die Analogie der Suche in Zeichenfolgen und Zeitreihen dargestellt. Darauf aufbauend sollen in den Abschnitten 4.2.2 und 4.2.3 die Möglichkeiten der Suche verbessert werden. In Abschnitt 4.2.4 soll eine Erweiterung der Suche in Arc Diagrams erfolgen, um auch merkmalübergreifend Muster finden zu können.

4.2.1 Analogie der Suche in Zeichenfolgen und Zeitreihen

Am Ende des dritten Kapitels wurde erwähnt, dass Arc Diagrams ganz allgemein Zeichenfolgen als Datenbasis verwenden. Diese Zeichenfolgen

werden auch *Strings* genannt. Es handelt sich dabei um geordnete Reihen von Symbolen mit einem bestimmten Informationsgehalt. Für die Mustersuche in vorliegender Zeitreihe kann eine Merkmalreihe allgemein als geordnete Zeichenfolge betrachtet werden. Der zeitliche Bezug in Merkmalreihen ist für eine Suche nach Mustern zunächst nicht von belang. Für eine Darstellung der Muster ist von Fall zu Fall zu überprüfen, ob der zeitliche Bezug im jeweiligen Verfahren adäquat veranschaulicht werden kann. Bestehende Analogien von Zeichenfolgen, einer multivariaten Zeitreihe und einer Merkmalreihe sind in Tabelle 4-1 dargestellt.

Tabelle 4-1: Analogien von Zeichenfolgen und Merkmalreihen

	Zeichenfolgen (Strings)	**Multivariate Zeitreihe**	**Merkmalreihe (univ. Zeitreihe)**
Auftretende Werte	*Symbole*	*Ausprägungen mehrerer Variablen*	*Ausprägungen einer Variablen*
Mögliche Werte	*Alphabetgröße*	*Umfang der Wertebereiche*	*Umfang des Wertebereichs*
Dimension	*1-dimensional*	*multivariat*	*univariat*
Strukturierung	*Sequentiell*	*Relational*	*Sequentiell*

Für die Suche nach Mustern ist es wichtig speziell auf einen Sachverhalt innerhalb der Tabelle 4-1 hinzuweisen. Zeichenfolgen sind 1-dimensional. Die vorliegende Arbeit untersucht aber multivariate Zeitreihen. Diese sind mehrdimensional (vgl. Tabelle 4-1). Um eine sinnvolle Analogie der Suche in Zeichenfolgen und multivariaten Zeitreihen zu erhalten, muss die Mustersuche in zwei Stufen zerlegt werden. Die erste Stufe sucht 1-dimensionale Muster innerhalb einer Merkmalreihe sukzessiv und getrennt für jede der Merkmalreihen der vorliegenden multivariaten Zeitreihe. Diese Stufe soll als univariate Mustersuche bezeichnet werden. Für die erste Stufe der Mustersuche in multivariaten Zeitreihen kann die Analogie zur Suche in Zeichenfolgen aufrechterhalten werden. Die zweite Stufe der Mustersuche soll mehrdimensionale Muster merkmalsübergreifend finden. Diese Stufe soll als multivariate Mustersuche bezeichnet werden. In Zeichenfolgen können solche mehrdimensionalen Muster nicht auftreten. Die Analogie zur Suche in Zeichenfolgen gilt hier nicht ohne weiteres.

4.2.2 Beschleunigung der Mustersuche

Die bestehende Analogie der Suche in Zeichenfolgen und der univariaten Suche wurde dargestellt. Der Algorithmus zur Mustersuche in Arc Diagrams ist für Zeichenfolgen konzipiert. Es wird jeweils nur eine einzige Zeichenfolge gleichzeitig betrachtet. Daher war es bislang nicht nötig, den Suchalgorithmus speziell auf seine Geschwindigkeit hin zu verbessern. In vorliegender multivariater Zeitreihe sollen zehn Merkmalreihen auf univariate Muster hin untersucht werden. Deshalb

muss die univariate Suche beschleunigt werden. Dazu sollen bereits anerkannte Algorithmen verwendet werden. Ziel ist es, univariate Muster aller Merkmalreihen in akzeptabler Zeit zu finden. Von akzeptabler Zeit kann dann gesprochen werden, wenn ein ausgewogenes Verhältnis zwischen den Kosten (Zeit zur Musterfindung) und Nutzen (Erkenntnisgewinn) der Darstellung bestehen bleibt. Diese Anforderung wurde im Abschnitt 2.1.1 allgemein als Angemessenheit eingeführt.

Die Verarbeitung langer Zeichenketten ist eine grundlegende Aufgabe in der Informatik. Bereits in den 70er Jahren wurden zahlreiche Probleme aus diesem Gebiet formuliert und auch effizient gelöst. Die Suche nach wiederkehrenden Mustern ist ein solches, bereits hinreichend gelöstes, Problem. Diese Tatsache ausnutzend sollen nun bestehende Algorithmen zur Suche in Zeichenketten verwendet werden, um univariate Muster auch in Merkmalen vorliegender Zeitreihe zu finden. Es ist an dieser Stelle zunächst erforderlich einige Begriffe und das „String-Matching-Problem" nach [La03] formal zu definieren.

Sei A ein Alphabet und seien $S = S_0,\ldots,S_{k-1}$ und $X_0,\ldots,X_{p-1}$ Zeichenfolgen der Länge k bzw. p über A.

Ein *Fenster* W_i ist eine Teilzeichenfolge von S der Länge p, die an Position i beginnt:

$W_i = S_i,\ldots,S_{i+p-1}$.

Ein Fenster W_i, das mit dem Muster X übereinstimmt, heißt *Vorkommen* des Musters an Position i:

W_i ist Vorkommen $\Leftrightarrow X = W_i$

Ein *Mismatch* in einem Fenster W_i ist eine Position j, an der das Muster mit dem Fenster nicht übereinstimmt.

j ist Mismatch in $W_i \Leftrightarrow X_j \neq (W_i)_j$.

String-Matching-Problem

Eingabe: Zeichenfolge $S = S_0,\ldots,S_{k-1}$ und Muster $X_0,\ldots,X_{p-1}$ über einem Alphabet A

Ausgabe: $\left\{ i \,\middle|\, X = W_i \right\}$

Oder Verbal: Es sind zwei Zeichenfolgen S und X gegeben. Gesucht sind alle Vorkommen des Musters X in der Zeichenfolge S.

Es existiert eine große Zahl an Verfahren, welche das beschriebene String-Matching-Problem lösen. Es ist zu untersuchen, welche Verfahren

für eine Steigerung der Geschwindigkeit der Mustersuche in Zeitreihen in Frage kommen. Ein geeignetes Kriterium zur Beschreibung dieser „Geschwindigkeit" ist die *Zeitkomplexität*. Sie soll an dieser Stelle kurz vorgestellt werden, um die später diskutierten Verfahren daraufhin vergleichen zu können. Eine formale Einführung erfolgt z.B. in [La03].

Die Anzahl der Schritte, die ein Algorithmus benötigt, wird als die *Laufzeit* des Algorithmus bezeichnet. Die Laufzeit eines Algorithmus ist in der Regel abhängig von der Länge der Eingabe *k*, d.h. der Größe des Problems, das der Algorithmus lösen soll. Die Anzahl der Schritte, die ein Algorithmus benötigt, wird daher immer als Funktion *T(k)* in Abhängigkeit von der Problemgröße *k* angegeben. Die Funktion *T(k)* heißt Zeitkomplexität des Algorithmus. Die Zeitkomplexität wird häufig in der O-Notation ausgedrückt. Um einen Algorithmus unabhängig von einer konkreten Eingabe bewerten zu können, gibt man für die Zeitkomplexität Folgendes an:

- Die maximale Anzahl benötigter Schritte des Algorithmus für eine beliebige Eingabe *k* (worst case)
- Die durchschnittliche Anzahl benötigter Schritte des Algorithmus für eine beliebige Eingabe *k* (average case)

Neben der, von einer konkreten Eingabe unabhängigen, Zeitkomplexität eines Algorithmus entscheidet ein wichtiges Kriterium über die Geschwindigkeit der Musterfindung: Die konkrete Eingabe, also die Eigenschaften der zugrunde liegenden Daten. Verfahren unterscheiden sich durch ihre Eignung für die Suche in z.B. kleinen Datenmengen, großen Alphabeten oder teilweise vorsortierten Daten. Eine geschlossene Lösung für beliebige Datenmengen und Aufgabenstellungen existiert nicht. Zunächst wird nun eine Betrachtung der konkreten Aufgabenstellung erfolgen, um eine Beschleunigung der Suche zu erreichen. Danach wird eine spezielle, an die Charakteristika der Daten angepasste, Verfahrensauswahl erfolgen.

Die allgemeine Aufgabenstellung ist es, alle Vorkommen des Musters *X* in einer Reihe *S* zu finden. Diese Aufgabenstellung soll speziell auf die Suche nach Mustern für Arc Diagrams übertragen werden, um daraus einen Nutzen abzuleiten. Die Suche nach essentiellen Paaren der Arc Diagrams weist zwei wesentliche Unterschiede zur allgemeinen Suche in Zeichenfolgen auf:

1. Das Muster *X* wird aus dem Text selbst extrahiert. Es existiert also mindestens ein Vorkommen des Musters in der Zeichenfolge. Für dieses Muster *X* wird ein korrespondierendes essentielles Paar *Y* gesucht.
2. Es ist nicht nach einem einzigen Muster *X* zu suchen. Die Arc Diagrams erfordern die Suche nach vielen Mustern mit unterschiedlichen Längen. Die mögliche Anzahl und Länge

der Muster wird bestimmt durch die Regeln für essentielle Paare der Arc Diagrams (vgl. Abschnitt 3.3).

Die Anzahl der benötigten Vergleiche der Mustersuche kann durch die genannten Unterschiede reduziert werden. Dies hat eine direkte Beschleunigung der Suche zur Folge.

- Es kommen für eine Merkmalreihe der Länge k nur Musterpaare der Länge $p \in \mathbb{N}$ mit $1 \leq p \leq \lceil (k-1)/2 \rceil$ in Betracht.

Der Grund dafür ist die Forderung „nicht-überlappend" der maximal passenden Paare in Verbindung mit der Tatsache, dass die Musterpaare aus der Merkmalreihe selbst gewonnen werden. Abbildung 4-1 zeigt ein essentielles Paar mit maximal möglicher Musterlänge. Die Länge der Merkmalreihe beträgt acht Ausprägungen. Die maximal mögliche Länge der Teilfolgen *X* und *Y* ist demnach vier Ausprägungen.

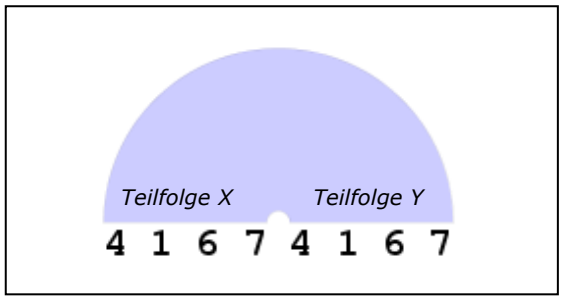

Abbildung 4-1: Ein essentielles Paar maximal möglicher Musterlänge

Eine weitere Reduktion der nötigen Vergleiche ist durch die Forderung „nicht-überlappend" im Zusammenhang mit der Forderung „fortlaufend" der maximal passenden Paare möglich:

- Bei der Suche nach einer Teilfolge *Y* reicht es aus, wenn für eine Suche von links nach rechts, im Intervall $[i+p, k-p]$ mit $p, k, i \in \mathbb{N}$ gesucht wird, *i* ist dabei die Position an der die Teilfolge *X* mit der Musterlänge *p* beginnt.

Ein Beispiel soll dies verdeutlichen. Für die betrachtete Teilfolge *X* mit der Musterlänge zwei und der Position eins in Abbildung 4-2, soll eine korrespondierende Teilfolge *Y* in der dargestellten Merkmalreihe gefunden werden. Mit dieser Suche kann an Position drei ($i+p$) begonnen werden. Diese „Vorwärtssuche" wird durch die Forderung „fortlaufend" der maximal passenden Paare ermöglicht. Die Forderung, dass sich Paare nicht überschneiden dürfen, ermöglicht schließlich die Suche ab der Position $(i+p)$. Beendet werden kann die Suche in Abbildung 4-2 nach der

Position sechs *(k-p)*. Die Suche an höhere Positionen ist nicht sinnvoll, da das Suchfenster W_i sonst über die Länge *k* der Merkmalreihe hinausreichen würde.

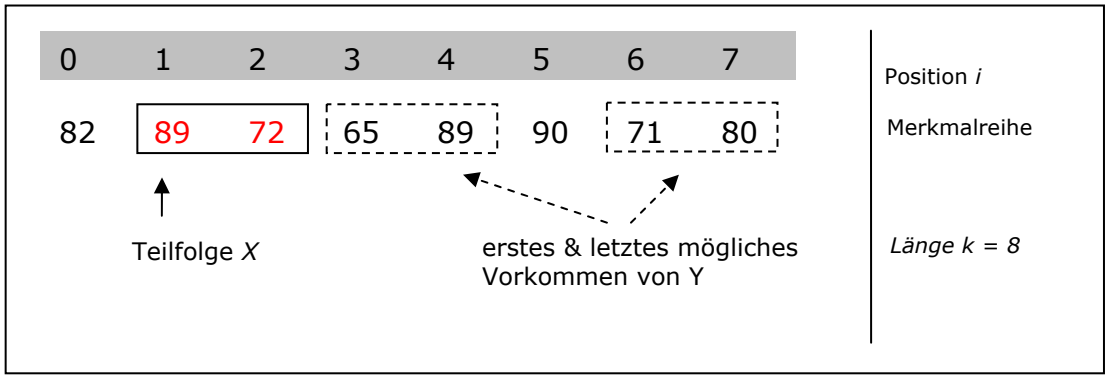

Abbildung 4-2: Mögliche Vorkommen von Teilfolgen *Y* bei gegebener Teilfolge *X*

Nach der genaueren Betrachtung der Suche speziell für Arc Diagrams, soll eine an die Charakteristika der Daten angepasste Auswahl von möglichen Verfahren zur Mustersuche erfolgen. Die Daten sind im Verlauf dieser Arbeit bereits als multivariate Zeitreihen charakterisiert worden. Bereits in Abschnitt 2.2.1 wurde erwähnt, dass der Umfang der untersuchten Zeitreihe bei etwa 40.000 Beobachtungszeitpunkten mit 10 Merkmalen pro Beobachtungspunkt liegt. Es sind also insgesamt etwa 400.000 Ausprägungen in der Zeitreihe vorhanden. Die Wertebereiche haben dabei merkmalsabhängig verschiedenen Umfang. Diese reichen von rund 50 bis etwa 500 möglichen Ausprägungen je Merkmal. Anstatt nur ein einzelnes anerkanntes Verfahren für eine univariate Mustersuche auszuwählen, geht diese Arbeit einen anderen Weg. Es sollen parallel drei Verfahren genutzt werden, um univariate Musterpaare zu finden. Diese Entscheidung beruht auf der Tatsache, dass nach [La03] für verschiedene Daten- und Mustereigenschaften einzelne Verfahren unterschiedlich gut geeignet sind. Die variierende Alphabetgröße in den einzelnen Merkmalen sei als nur ein Beispiel genannt. Es wird also nicht ein einzelnes Verfahren verwendet werden, in der Annahme, dass dieses in allen Merkmalen und für alle univariaten Muster effizient sucht. Es wird fallbezogen die Auswahl zwischen drei Verfahren erfolgen:

- der Naiven Suche,
- der Wahrscheinlichkeitsgestützten Suche und
- der Suche nach dem Karp-Rabin Algorithmus.

Bevor diskutiert wird, wann welches Verfahren zum Einsatz kommt, werden die Verfahren vorgestellt.

Die Naive Suche

Dies ist der einfachste Algorithmus zur Suche eines Musters. Er überprüft das Muster an allen Positionen i=0, …,k-p. Es wurde gezeigt, dass für die konkrete Problemstellung zur Suche nach essentiell passenden Paaren eine Suche von i=i+k,…,k-p ausreicht. Das Muster wird an der jeweiligen Position Ausprägung für Ausprägung von links nach rechts mit dem Merkmalsfenster verglichen. Bei einem Mismatch oder vollständiger Übereinstimmung der Ausprägungen wird das Muster um eine Position weiter geschoben.

Für die Anzahl der nötigen Vergleiche v für den Naiven Algorithmus zur Suche nach möglichen Teilfolgen Y gilt:

$$v \leq (k - p - p + 1) \cdot p \quad \Rightarrow \quad v \in O(k \cdot p).$$

Im schlechtesten Fall sind also genau $(k - p - p + 1) \cdot p$ Vergleiche v erforderlich. Es kann gezeigt werden, dass für die minimale Anzahl der Vergleiche die untere Schranke $v \in \Omega(k \cdot p)$ gilt. Dies bedeutet, dass die Naive Suche mindestens aber auch höchstens proportional zu $(k \cdot p)$ viele Schritte benötigt.

Der fundamentale Unterschied des Naiven Algorithmus zu anderen Suchverfahren ist, dass kein so genanntes *Preprocessing* erfolgt. Dies ist eine Vorlaufphase der Mustersuche, um Vorab-Informationen über die Struktur eines Musters und die darin enthaltenen Ausprägungen zu gewinnen. Mit Hilfe dieser Informationen sollen dann möglichst weniger Vergleiche zur Suche nach einem Muster erfolgen. Dies hat eine bessere Zeitkomplexität des Algorithmus in der Phase des Suchens zur Folge. Der Algorithmus ist dann „schneller". Das fehlende Preprocessing ist auch der Grund für die Anzahl der nötigen Vergleiche einer Suche im schlechtesten Fall. Wann es dennoch sinnvoll ist, dieses Verfahren einzusetzen, wird nach Vorstellung der beiden übrigen Verfahren diskutiert werden.

Die Wahrscheinlichkeitsgestützte Suche

Es ist prinzipiell nicht nötig, die einzelnen Ausprägungen einer gesuchten Teilfolge Y von links nach rechts mit den Ausprägungen der Teilfolge X zu vergleichen. Sie können theoretisch in beliebiger Reihenfolge verglichen werden, um benötigte Vergleiche bei der Suche einzusparen.

Die Idee der Wahrscheinlichkeitsgestützten Suche ist, die Ausprägungen entsprechend ihrer Auftretenswahrscheinlichkeit zu vergleichen. Ausprägungen, welche möglichst selten in der Merkmalreihe vorkommen, werden zuerst verglichen. Diese verursachen mit der größten Wahrscheinlichkeit einen Mismatch, sodass das Fenster zur Suche weiter geschoben werden kann. Es wird vorausgesetzt, dass die Häufigkeitsverteilung der Ausprägungen bekannt ist. Ermittelt wird diese Verteilung im Preprocessing. Im Prinzip müssen die Ausprägungen eines Merkmals dazu nach ihrer Auftretenswahrscheinlichkeit aufsteigend

sortiert werden. Es gibt Sortieralgorithmen die diesen Schritt im schlechtesten Fall in der Zeitkomplexität $T(k) \in O(k \cdot log(k))$ für eine gegebene Länge k einer Zeichenfolge erledigen. Das Vorgehen der Wahrscheinlichkeitsgestützten Suche soll an einem Beispiel demonstriert werden. In Abbildung 4-3 tritt die Ausprägung „a" häufiger auf als die Ausprägung „b". Da eine Ausprägung „b" in der Teilfolge *X* (aba) enthalten ist, wird diese zuerst verglichen. In der Abbildung 4-3 führt so schon der jeweils erste Vergleich ab der Position drei und vier zu einem Mismatch. Dies muss nicht immer so sein. Man kann mit diesem Algorithmus die Wahrscheinlichkeit dazu jedoch erhöhen. Ab der Position fünf werden drei Vergleiche durchgeführt. An dieser Stelle wird ein Vorkommen von *X*, die korrespondierende Teilfolge *Y*, gefunden.

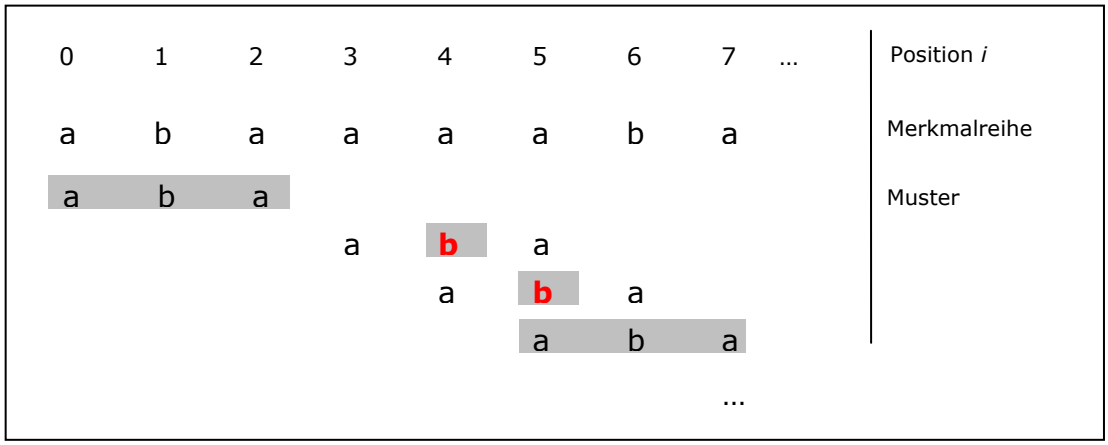

Abbildung 4-3: Das Prinzip der Suche des Wahrscheinlichkeitsgestützten Algorithmus

Wie viele Vergleiche sich mit Hilfe des Wahrscheinlichkeitsgestützten Algorithmus einsparen lassen, hängt von der Wahrscheinlichkeitsverteilung der Ausprägungen eines Merkmals ab. Im schlechtesten Fall gar keine. Dies ist genau dann der Fall, wenn alle Ausprägungen gleichwahrscheinlich auftreten. Dann hat dieser Algorithmus dieselbe Zeitkomplexität wie die Naive Suche. Kommt jedoch in der Teilfolge *X* eine Ausprägung vor, dessen Auftretenswahrscheinlichkeit sehr gering ist, dann ist dieser Algorithmus gegenüber der Naiven Suche deutlich im Vorteil.

Der Karp-Rabin Algorithmus

Der dritte Suchalgorithmus, welcher für das vorgestellte Konzept Verwendung finden soll, ist der Karp-Rabin Algorithmus. In gewisser Weise ähnelt das Verfahren von Karp und Rabin [KR87] dem Naiven und dem Wahrscheinlichkeitsgestützten Algorithmus. Das Muster *X* wird nach einem Mismatch oder einer vollständigen Übereinstimmung ebenfalls um eine einzige Position weiter geschoben. Es wird also mit denselben Fenstern W_i verglichen. Anstatt jedoch die einzelnen Ausprägungen in

einer bestimmten Reihenfolge zu vergleichen, wird nur ein einziger Vergleich durchgeführt. Die *Signatur* des Musters wird mit der Signatur des Fensters verglichen. Eine Signatur ist ein möglichst eindeutiges Kennzeichen des Musters bzw. des Fensters. In der Regel handelt es sich um einen Wert, welcher durch eine bestimmte *Signaturfunktion* im Preprocessing ermittelt wurde. Die formalen Definitionen lauten wie folgt [La07]:

> Sei A ein Alphabet und M eine Menge. Eine Signaturfunktion ist eine Abbildung $f : A^* \to M$, die jedem Muster bzw. Fenster $(X, W_i) \in A^*$ einen Wert aus $f(X) \in M$ bzw. $f(W_i) \in M$ zuordnet. Der Wert $f(X)$ bzw. $f(W_i)$ ist die Signatur.

Wenn die Signatur eines Musters mit der eines Fensters nicht übereinstimmt, so handelt es sich garantiert um kein Vorkommen an dieser Position. Wenn allerdings die Signatur übereinstimmt, kann es sich um ein Vorkommen handeln. Durch die Verwendung von Signaturen liefert der Karp-Rabin Algorithmus eine Menge von Kandidaten von Vorkommen des zu suchenden Musters. Damit das Verfahren möglichst effizient arbeitet, sind an die Signaturfunktion zwei Anforderungen zu stellen:

- *Verwechslungen* sollten möglichst ausgeschlossen sein.
- Die Signatur muss sich in konstanter Zeit berechnen lassen.

Eine Verwechslung tritt genau dann ein, wenn die Signatur des Musters und des Fensters übereinstimmt, tatsächlich aber kein Vorkommen vorliegt. Formal ausgedrückt nach [La07]:

> Sei $f : A^* \to M$ eine Signaturfunktion.
> Eine Verwechslung ist ein Paar von Teilfolgen (X, W_i) mit
> $X \neq W_i$ aber $f(X) = f(W_i)$

Es wird deutlich, dass die Wahl der Signaturfunktion einen entscheidenden Einfluss auf die Effizienz des Karp-Rabin Algorithmus hat. Dies soll am Beispiel der Quersumme als Signaturfunktion in Abbildung 4-4 gezeigt werden. Die Anzahl der möglichen Ausprägungen der Merkmalreihe ist zehn. Die aus der Merkmalreihe extrahierte gegebene Teilfolge X sei „7 6". Die Quersumme von X ist 7+6=13. Anhand dieser Quersumme soll die zugehörige Teilfolge Y gesucht werden. Gemäß den erarbeiteten Vorüberlegungen wird an der Position zwei mit der Suche begonnen. Die Quersumme des Fensters W_i an dieser Stelle ist neun. Es kann sich also nicht um ein Vorkommen handeln. Bei der Quersumme ab Position vier stimmen die Quersummen überein. Es wird nun getestet, ob es sich tatsächlich um ein Vorkommen handelt, oder ob eine Verwechslung vorliegt. Aus der Abbildung 4-4 ist ersichtlich, dass es sich beim Fenster W_i ab der Position vier um eine Verwechslung handelt.

In der Tat ist die Quersumme als Signaturfunktion gemäß den formulierten Anforderungen nicht geeignet, da eine hohe Anfälligkeit für Verwechslungen beobachtet werden kann. Diese Verwechslungen erfordern Vergleiche, welche Zeit kosten, ohne das dabei tatsächlich ein Muster gefunden werden kann. Ein Vorteil der Quersumme ist allerdings, dass sich jede Signatur aus der vorhergehenden in konstanter Zeit berechnen lässt. Die Ausprägung die das Fenster W_i verlässt, wird dazu von der Quersumme subtrahiert und die neu hinzukommende Ausprägung addiert.

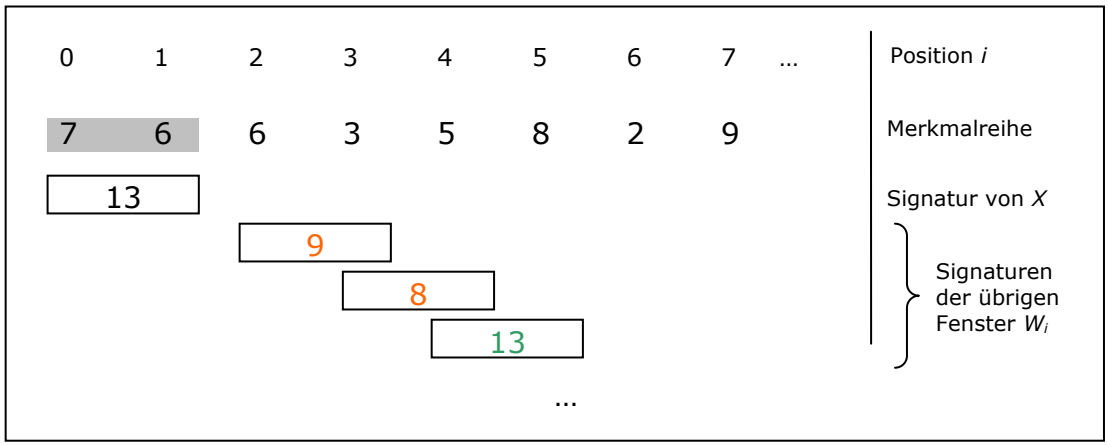

Abbildung 4-4: Quersummen als Signaturfunktion des Karp-Rabin Algorithmus in verschiedenen Textfenstern

Für die Verwendung des Karp-Rabin Algorithmus in dieser Arbeit wird die Signaturfunktion nach der *Divisionsrestmethode* gewählt. Die Funktion lautet dann $f(W_i)=W_i \bmod q$. Zwei Signaturen stimmen also dann überein, wenn sie bei Division durch den Modul q denselben Rest haben. Im Prinzip ist die Division mit Rest nicht auf ganze Zahlen beschränkt. In dieser Arbeit wird aber, aufgrund der einfacheren Berechnungen, eine solche Beschränkung empfohlen. Liegen nicht alle Ausprägungen einer Merkmalreihe im Bereich der natürlichen Zahlen, werden die möglichen Ausprägungen dazu von 1 bis zum Umfang des Wertebereiches durchnumeriert. Die so erhaltene Ziffernfolge soll als Zahl interpretiert werden. Durch dieses Vorgehen wird die Divisionsrestmethode sehr flexibel. Ausprägungen z.B. aus dem Bereich der reellen Zahlen oder Zeichenfolgen können jetzt ebenfalls verarbeitet werden. Es kann gezeigt werden, dass es günstig ist, für q eine Primzahl zu wählen. Weiterhin sollte q groß genug sein, um Verwechslungen möglichst zu vermeiden. Zur optimalen Größe von q sind kaum konkrete Angaben in der Literatur zu finden. Es soll zumindest $q \gg p$ gewählt werden.

Es sei erwähnt, dass Divisionen zunächst zeitaufwendiger als z.B. die Additionen der Quersummensignaturfunktion sind. In einer Implementierung kann jedoch eine Hashtabelle zur Abbildung dieser Divisionsrestmethode verwendet werden, wodurch in großen

Datenmengen ein schneller Zugriff und ein schneller Vergleich der gespeicherten Ausprägungen ermöglicht wird.

Gemäß den Anforderungen an eine Signaturfunktion ist die Divisionsrestmethode gut geeignet: Die Signatur eines Wertes kann in konstanter Zeit aus der vorherigen Signatur berechnet werden. Bei geeigneter Wahl von *q* treten nur sehr selten Verwechslungen auf.

Zur Zeitkomplexität des Karp-Rabin Algorithmus ist folgendes festzuhalten. Im Preprocessing erfolgt die Berechnung der Signaturen. Die Zeitkomplexität des Preprocessing liegt in *O(p).* Ein Nachteil des Karp-Rabin Algorithmus ist seine Zeitkomplexität im schlechtesten Fall des Suchens. Sie ist identisch mit dem Naiven Algorithmus: $T(k) \in O(k \cdot p)$. Seine durchschnittliche Laufzeit beträgt allerdings $T(k) \in O(k)$. Deswegen soll der Algorithmus, neben dem Naiven und dem Wahrscheinlichkeitsgestützten Ansatz, für eine Beschleunigung der Suche von essentiell passenden Paaren verwendet werden.

Fallbezogene Auswahl der vorgestellten Verfahren

Drei Verfahren für eine angestrebte Beschleunigung der univariaten Mustersuche in Arc Diagrams wurden vorgestellt. Es erfolgt nun eine fallbezogene Auswahl zwischen den drei Verfahren. Diese Auswahl beruht auf den erarbeiteten Vor- und Nachteilen, sowie auf der Berücksichtigung der jeweiligen Daten- und Mustereigenschaften der Merkmale und ihrer Ausprägungen. Die abstrakte Beschreibung des Vorgehens zur Verfahrensauswahl lautet wie folgt:

Eingabe: Merkmalreihe $S = S[0,..,k]$
WIEDERHOLE für alle möglichen Teilfolgenlängen *p*
  WIEDERHOLE für alle möglichen Muster *X* der Länge *p*
    Muster *X* aus *S* extrahieren
    WIEDERHOLE Für alle möglichen Teilfolgenpositionen *Y*
    FALLS ($p \leq 3$) DANN **Naive Suche**
    SONST FALLS (Wahrscheinlichkeitstest(*X*))
      DANN **Wahrscheinlichkeitsgestützte Suche**
      SONST **Karp-Rabin-Suche**

Diese abstrakte Beschreibung soll näher erläutert werden. Auf diesem Weg wird auch der positive Nutzen für die Beschleunigung der univariaten Suche aufgezeigt. Als Diskussionsgrundlage sollen zunächst exemplarisch einige der Anzahlen unterschiedlichen Musterlängen in vorliegender multivariater Zeitreihe ermittelt werden. Dazu zeigt Abbildung 4-5 die Wertetabelle der ermittelten Musterlängen dreier Merkmale einer Stichprobe von 1000 Ausprägungen. Die rechte Seite der Abbildung 4-5 zeigt ein Säulendiagramm, in welchem die Anzahl der Musterlängen in Abhängigkeit des Umfangs des Wertebereichs der

Merkmale abgetragen ist. Die Y-Achse des Säulendiagramms ist logarithmisch skaliert.

	Wolken-bedeckung [%]	Luft-feuchte [%]	Luftdruck [hPa]
Umfang Werte-bereich	*31*	*67*	*345*
p=1	*969*	*933*	*655*
p=2	*494*	*224*	*9*
p=3	*82*	*8*	*0*
p>3	*17*	*0*	*0*

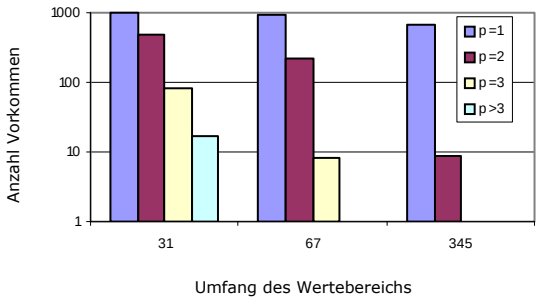

Abbildung 4-5: Darstellung der Anzahl an Vorkommen verschiedener Musterlängen *p* in Abhängigkeit des Wertebereichs dreier Merkmale für eine Stichprobe von 1000 Ausprägungen

Die Darstellung und die Wertetabelle zeigen, dass die Anzahl der auftretenden Muster direkt vom Umfang des Wertebereichs abhängt. Diese Tatsache soll in 4.2.3 aufgegriffen und zunächst nicht weiter betrachtet werden.

In allen Merkmalreihen treten die Musterpaare mit der Länge p=1 bei weitem am häufigsten auf. Für eine Beschleunigung ist die Wahl eines geeigneten Verfahrens zur Suche nach Musterpaaren der Länge p=1 daher sehr wichtig. Für diese Suche wird die Verwendung des Naiven Algorithmus vorgeschlagen. Es wurde dargestellt, dass sein Zeitverhalten für den schlechtesten Fall $T(k) \in O(k \cdot p)$ beträgt. Für p=1 wird daraus $T(k) \in O(k)$. Diese Zeitkomplexität wird auch von den besten String-Matching Algorithmen erreicht (vgl. [La03]). Der Naive Algorithmus benötigt dazu jedoch kein Preprocessing. Im Ergebnis heißt das, der Naive Algorithmus ist bei der Mustersuche nach Paaren der Länge p=1 schneller als jedes untersuchte Verfahren.

Im Verlauf der weiteren Untersuchungen zeigte sich, dass der Naive Algorithmus allgemein für kleine Musterlängen praktikabel arbeitet. Es ist zwar möglich durch die Wahl anderer Verfahren Vergleiche einzusparen. Diese Einsparung ist jedoch mit einem Preprocessing verbunden, welches Zeit erfordert. Es wird daher vorgeschlagen den Naiven Algorithmus für Musterpaare bis zu einer Länge $p \leq 3$ zu benutzen. Soll nach längeren essentiellen Paaren gesucht werden, sind andere Verfahren in der Regel schneller. Merkmalsabhängig ist also etwa ab dieser Grenze der Mehraufwand durch eine „intelligentere" Suche mit Hilfe eines Preprocessing für einen Geschwindigkeitszuwachs lohnend.

Auch wenn die Abbildung 4-5 verdeutlicht, dass die Chance zum Finden von Mustern der Länge $p>3$ in Abhängigkeit der Größe des Wertebereichsumfangs recht gering ist, so ist eine Suche nach diesen „längeren" Musterpaaren dennoch wichtig. Wird ein solches Musterpaar gefunden, ist seine Bedeutung für das Verständnis struktureller Zusammenhänge in der Merkmalreihe wesentlich.

Für die Suche nach essentiell passenden Paaren einer Länge $p>3$ wird in dieser Arbeit der Einsatz der Wahrscheinlichkeitsgestützten Suche und der Suche nach dem Karp-Rabin Algorithmus vorgeschlagen. Die Auswahl zwischen diesen beiden Verfahren soll merkmalsabhängig erfolgen. Dies soll am Beispiel der Abbildung 4-6 verdeutlicht werden. Die Abbildung zeigt ein Histogramm des bereits in Abbildung 4-5 erwähnten Merkmals „Luftfeuchte".

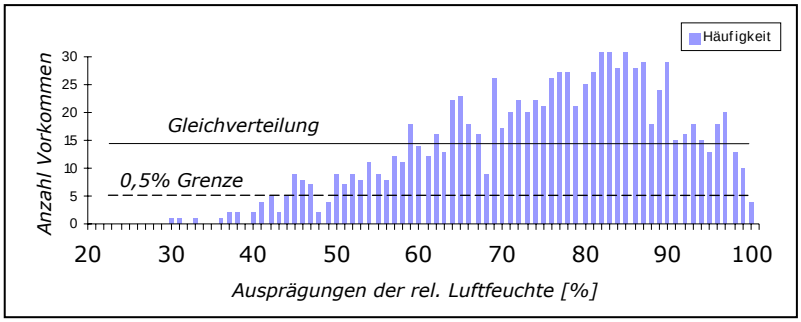

Abbildung 4-6: Histogramm über das Merkmal „Luftfeuchte" (Stichprobe von 1000 Ausprägungen)

In Abbildung 4-6 ist mit einer durchgehenden Linie der Wert von 15 Vorkommen gekennzeichnet. Diese Anzahl müssten alle Ausprägungen etwa annehmen, wenn sie gleichwahrscheinlich aufträten. Die Abbildung zeigt, dass dies nicht der Fall ist. Folglich soll für Musterlängen $p>3$ der Wahrscheinlichkeitsgestützte Suchalgorithmus verwendet werden. Auf diese Weise können etwa 20% der Vergleiche gegenüber dem Naiven Algorithmus eingespart werden. Ein Nachteil ist jedoch, dass das Sortieren der gesamten Ausprägungen im Preprocessing viel Zeit erfordert. Die Beschleunigung der Suche fällt deshalb viel geringer als 20% gegenüber dem Naiven Algorithmus aus.

Für das Preprocessing wird daher folgende Variante vorgeschlagen. Es wird ein Wahrscheinlichkeitstest durchgeführt. Dieser leistet folgendes: Es werden nicht alle Ausprägungen sortiert. Es wird lediglich ermittelt, ob in der zu suchenden Teilfolge Y Ausprägungen vorhanden sind, die eine Auftretenswahrscheinlichkeit von kleiner gleich 0,5% besitzen. Ist dies der Fall, wird diese Ausprägung zuerst verglichen. Der Rest der Werte wird nicht sortiert. In Abbildung 4-6 ist die untere Grenze von 0,5% zur Suche mit dem Wahrscheinlichkeitsgestützten Verfahren eingetragen. Es kann abgelesen werden, dass 14 der 67 Ausprägungen eine Auftretenswahrscheinlichkeit von kleiner gleich 0,5% besitzen. Kommt eine dieser Ausprägungen, in der zu suchenden Teilfolge X vor, so wird

die Wahrscheinlichkeitsgestützte Suche zur Verwendung vorgeschlagen. Kommt keine solche Ausprägung vor, dann wird der Karp-Rabin Algorithmus für die Suche verwendet.

Zusammenfassung und Fazit

Ziel dieses Abschnitts war die Beschleunigung der univariaten Mustersuche, um diese Muster in mehr als einer Merkmalreihe in akzeptabler Zeit zu finden. Dieses Ziel wurde durch zwei unterschiedliche Ansätze erreicht. Zunächst wurde das allgemeine Suchproblem für Zeichenfolgen auf die Suche für Arc Diagrams spezialisiert. Dies hat eine Reduktion der nötigen Vergleiche und damit eine Zeitersparnis zur Folge. Weiterhin wurden drei String-Matching Verfahren diskutiert. Anstatt ein einzelnes Verfahren für die Beschleunigung der univariaten Mustersuche auszuwählen, wurden die drei Algorithmen musterabhängig kombiniert. Diese fallbezogene Auswahl ermöglicht eine zusätzliche Beschleunigung der Suche.

Das vorgestellte Beschleunigungskonzept ist vor dem Hintergrund multivariater Zeitreihen entworfen. Abschließend sei erwähnt, dass es im Grunde aber nicht auf multivariate Zeitreihen beschränkt ist. Stattdessen ist es beliebig für die univariate Suche nach Musterpaaren der Arc Diagrams einsetzbar. Dabei spielt es keine Rolle, ob das Beschleunigungskonzept für die univariate Suche bezüglich nur eines Merkmals oder für viele Merkmale nacheinander verwendet wird. Weiterhin werden keine besonderen Anforderungen an den Wertebereich der Merkmale gestellt. Ein nominales Skalenniveau der Merkmale ist ausreichend. Somit ist das vorgestellte Beschleunigungskonzept sehr flexibel einsetzbar.

4.2.3 Steigerung der Flexibilität der Mustersuche

Im vorigen Abschnitt wurde betrachtet, wie man die univariate Suche nach essentiell passenden Paaren beschleunigen kann. Dabei wurde davon ausgegangen, dass die Teilfolge *X* und die zu findende Teilfolge *Y* exakt übereinstimmen. Aus zwei Gründen soll eine Erweiterung der exakten Suche vorgenommen werden:

1. Der Umfang des Wertebereichs ist im Verhältnis zur Merkmalreihenlänge *k* sehr groß. In Abschnitt 4.2.2 wurde deutlich, dass der Umfang des Wertebereichs einen direkten Einfluss auf das Finden von Musterpaaren hat (vgl. auch Abbildung 4-5). Es ist möglich die Wahrscheinlichkeit zur Findung einer korrespondierenden Teilfolge *Y* zu steigern, indem man gewisse Toleranzen für *Y* zulässt.
2. Mögliche Muster sollen auch in fehlerbehafteten Merkmalreihen gefunden werden. Zeitreihen werden häufig mit Hilfe von Sensoren erfasst. In der Biometrie werden Zeitreihen von Messungen an Lebewesen durch Sensoren

erstellt. Die Meteorologie verwendet Sensoren, um Zeitreihen verschiedener Grundgrößen aufzunehmen. Allen Sensoren haben gemeinsam, dass mit Ihnen ein fehlerfreies Messen unmöglich ist. Die Größe des Fehlers kann je nach Sensor und Messbedingungen stark variieren. Entscheidend ist, dass Ausprägungen aufgenommen werden, die nicht mit den wahren Ausprägungen der Merkmale übereinstimmen.

Eine flexiblere Mustersuche soll die zwei genannten Fälle geeignet behandeln. Muster sollen jetzt als Vorkommen erkannt werden, wenn sie eine *hinreichende Ähnlichkeit* besitzen. Als hinreichend ähnlich sollen solche Musterpaare bezeichnet werden, die innerhalb bestimmter Toleranzen übereinstimmen. Im Zuge dieser Arbeit sollen zwei Ansätze vorgestellt werden, um diese Toleranzen festzulegen:

- Die Unscharfe Suche und
- die Distanzsuche.

Die formale Definition für ein *Vorkommen* wird sich dafür wie folgt ändern (vgl. Abschnitt 4.2.2):

> Ein Fenster W_i, das mit dem Muster X hinreichend ähnlich ist, heißt Vorkommen des Musters an Position i:
>
> W_i ist Vorkommen $\Leftrightarrow X \cong W_i$

Die Unscharfe Suche

Die Idee der Unscharfen Suche ist, Intervalle um die Ausprägungen eines Merkmals zu legen. Diese bestimmen den Toleranzbereich für eine hinreichende Ähnlichkeit. Liegen die zu vergleichenden Ausprägungen also innerhalb dieser Intervalle, wird ein Vorkommen erkannt. Zur Festlegung dieser Intervalle werden metrische Skalenniveaus vorausgesetzt. Im Abschnitt 2.2.1 wurde dargestellt, dass diese Forderung für die vorliegenden Daten erfüllt ist. Es wird vorgeschlagen die Größe der Intervalle in Abhängigkeit der Größe des Umfangs des Wertebereichs jedes Merkmals auf folgende Weise zu bestimmen:

1. Spannweite r des Wertebereichs eines Merkmals festlegen:
   r = max. Ausprägung – min. Ausprägung
2. Unschärfefaktor u festlegen mit $u \in (0,...,1]$; $u \in \mathbb{R}$.
3. Intervall für jede Ausprägung bestimmen nach:
   $\left[Ausprägung - |r| \cdot u, Ausprägung + |r| \cdot u \right]$

Die Bestimmung des individuellen Intervalls für jede Ausprägung ist zeitaufwendig. Dennoch ist dieses Vorgehen sinnvoll, weil dadurch eine

Anpassung des Intervalls an die Größenordnung der Ausprägungen und der Merkmale erfolgt. Der Wert des Unschärfefaktors *u* bestimmt die Spannweite *r* des symmetrischen Intervalls um die Ausprägung. Für *u*=0 wäre die Unscharfe Suche identisch mit der Exakten Suche. Für *u*=1 ist das Intervall um jede Ausprägung so groß, dass garantiert alle anderen Ausprägungen als korrespondierende Teilfolge *Y* dieser Ausprägung erkannt werden. Die Wahl eines geeigneten Faktors *u* ist also sehr wichtig für den Erfolg der unscharfen Suche.

Die unscharfe Suche soll am Beispiel der Abbildung 4-7 veranschaulicht werden. Zuerst wird die Spannweite des Wertebereichs der Merkmalreihe bestimmt. Diese beträgt *r*=10.0 (15.0-5.0). Zweitens wird der Unschärfefaktor festgelegt. Der Unschärfefaktor betrage *u*=0.5. Die aus der Merkmalreihe extrahierte Teilfolge *X* sei „10.0 8.0 5.0". Drittens werden nun die Intervalle für jede auftretende Ausprägung in *X* gebildet. Grün sind in der Abbildung die Ausprägungen eingetragen, die innerhalb der gebildeten Intervalle liegen. Die rote Ausprägung zeigt, dass die Bedingung für ein Vorkommen nicht erfüllt wird.

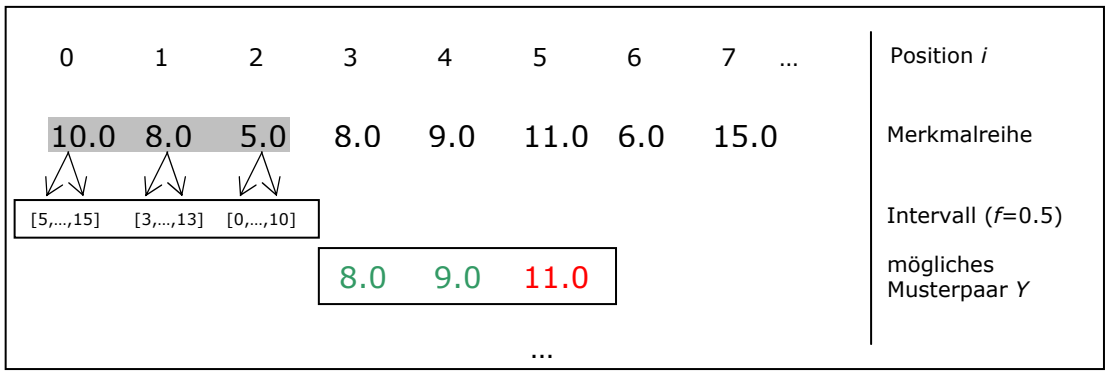

Abbildung 4-7: Das Prinzip der Unscharfen Suche am Beispiel eines Musters mit der Länge *p*=3

Ausprägungen die im Bereich der reellen Zahlen $\mathbb{R}$ liegen, haben häufig einen großen Umfang des Wertebereichs. Mit Hilfe der Unscharfen Suche können Musterpaare auch in Merkmalreihen mit einem erhöhten Umfang des Wertebereichs gefunden werden. Steuerbar ist dieser Prozess des Findens durch den Unschärfefaktor *u*.

Weiterhin ist die Unscharfe Suche gut geeignet, um Abweichungen durch Messfehler in den Merkmalreihen auszugleichen. Der Messfehler sollte dabei im Verhältnis zur Ausprägung in allen Merkmalreihen etwa die gleiche Größenordnung haben. Sollen aber *grobe Fehler* in den Merkmalreihen berücksichtigt werden, ist die Unscharfe Suche wenig geeignet. Grobe Fehler seien einzelne gemessene Ausprägungen, welche überdurchschnittlich stark von der wahren Ausprägung abweichen. Sie entstehen überwiegend durch die falsche Bedienung der Messinstrumente. Um eine Möglichkeit zur Berücksichtigung grober Fehler zur Verfügung zu haben, soll nun die Distanzsuche vorgestellt werden. Ihr Ziel ist es zusammenhängende Muster in Teilfolgen zu

erkennen, auch wenn einzelne Ausprägungen sich durch grobe Fehler sehr stark unterscheiden.

Die Distanzsuche

Die Distanzsuche legt als Maß für eine hinreichende Ähnlichkeit eine Distanz zwischen Teilfolgen fest. Hinreichend ähnlich sind Teilfolgen dann, wenn eine festgelegte obere Grenze der Distanz nicht überschritten wird. Zur Bestimmung dieser Distanz sind verschiedene Operationen zulässig. Die verschiedenen Operationen haben die Aufgabe eine Teilfolge *X* in eine Teilfolge *Y* zu überführen. Die Anzahl der nötigen Operationen bestimmt die Distanz zwischen beiden Musterpaaren. Dabei setzt das Zählen der ungleichen Ausprägungen in Teilfolgen lediglich ein nominales Skalenniveau der Daten voraus (vgl. Abschnitt 2.2.1). Es wird zwischen Gleichheit oder Ungleichheit von Ausprägungen unterschieden. Aus diesem Grund ist die Distanzsuche zur Berücksichtigung grober Fehler bei der Mustererkennung gut geeignet. Die Größenordnung des Fehlers spielt keine Rolle. Es gibt verschiedene Distanzfunktionen. Diese unterscheiden sich je nach Anwendungsfeld und zulässigen Operationen.

In dieser Arbeit sollen nur Musterpaare gefunden werden, die in ihrer Länge übereinstimmen. Für eine Distanzfunktion reicht es demnach aus, wenn eine Operation „Ersetzung" zur Verfügung steht. Diese Tatsache führt zur Hamming-Distanz. Die Hamming-Distanz ist ursprünglich ein Maß für die Unterschiedlichkeit digitaler Daten. Im Zuge dieser Arbeit soll die Hamming-Distanz als die minimal benötigte Anzahl der Operation „Ersetzung" verstanden werden, um die Teilfolge *Y* in die Teilfolge *X* zu überführen.

Auf Basis der Hamming-Distanz wird ein Toleranzbereich für die flexible Suche nach Musterpaaren in Zeitreihen vorgeschlagen. Die Anzahl an nicht übereinstimmenden Ausprägungen eines Musters sei die Fehleranzahl *anzF*. Die Anzahl an Übereinstimmungen *ü* von Ausprägungen aus *X* und *Y* für eine gegebene Musterlänge *p* ist demnach:

$$ü = p - anzF; \quad ü \geq 0; \quad p \geq anzF; \quad ü, p, anzF \in \mathbb{N}$$

Da die Suche nach essentiellen Paaren viele unterschiedliche Musterlängen einschließt, ist es nicht sinnvoll eine absolute Grenze für Übereinstimmungen anzugeben. Deshalb erfolgt eine Normierung auf die Musterlänge:

$$ü_N = \frac{p - anzF}{p} \qquad \Rightarrow \qquad ü_N = 1 - \frac{anzF}{p}; \quad ü_N \in [0,...,1]; \; ü_N \in \mathbb{R}$$

Die Wahl eines geeigneten Toleranzbereiches für $ü_N$ ist abhängig von den vorliegenden Daten. In dieser Arbeit wird aufgrund der untersuchten Zeitreihe ein Toleranzbereich von $1 \geq ü_N \geq 0{,}75$ vorgeschlagen.

Abbildung 4-8 zeigt eine Arc Diagram Darstellung nach der vorgeschlagenen Distanzsuche (rechts) im Vergleich zur exakten Suche (links). Ein Fehler ist in der Merkmalreihe in roter Farbe dargestellt. Es zeigt sich, dass dieser Fehler die Anzeige eines Vorkommens der Länge vier auf der linken Seite verhindert. Die Distanzsuche auf der rechten Seite der Abbildung findet das Vorkommen.

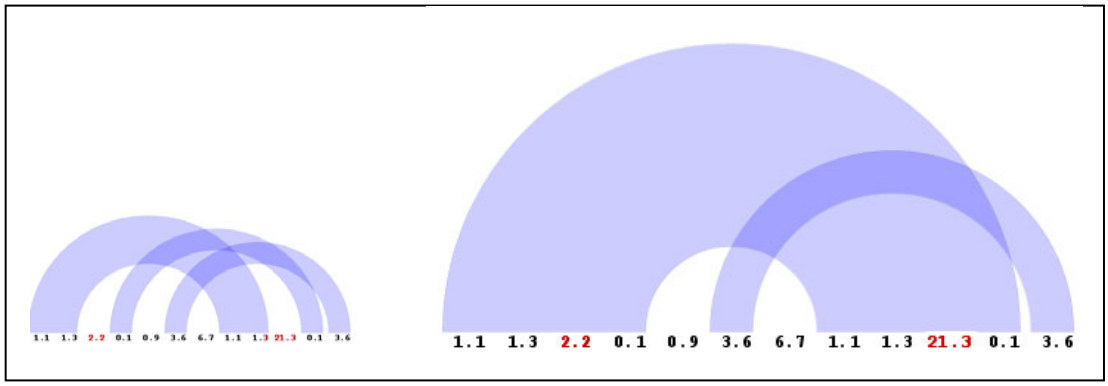

Abbildung 4-8: Arc Diagram Darstellung einer Merkmalreihe nach exakte Suche (links) und der Distanzsuche für $T \geq 0{,}75$ (rechts)

Zusammenfassung und Fazit

Durch die Distanzsuche und die Unscharfe Suche ist es möglich den Einfluss eines großen Wertebereichsumfangs auf die Musterfindung zu reduzieren, um überhaupt zusammenhängende Musterpaare zu finden.

Weiterhin können beide Verfahren gut geeignet sein, um Muster in Merkmalreihen finden zu können, in denen einzelne Ausprägungen Fehler aufweisen. Bei der Distanzsuche spielt die Größenordnung des Fehlers dabei keine Rolle. So kommt die Distanzsuche auch für nicht-numerische Daten in Betracht. Im Unterschied dazu setzt die Unscharfe Suche die Größenordnung des Fehlers ins Verhältnis zur Spannweite des Wertebereichs eines Merkmals. Die Unscharfe Suche ist damit nur für Daten mit einem metrischen Skalenniveau verwendbar.

Für beide Verfahren ist es wichtig, den Toleranzbereich zur Erkennung von Vorkommen geeignet festzulegen. Ein zu großer Toleranzbereich erkennt zu viele Musterpaare. Wird der Toleranzbereich zu klein gewählt, hat die Flexiblere Suche keinen zusätzlichen Effekt gegenüber der Exakten Suche.

Für die weiteren Untersuchungen wird es noch von Bedeutung sein, dass mit zunehmendem Toleranzbereich im Allgemeinen auch die Länge der erkannten Muster zunimmt. Diese Tatsache soll in der Darstellung von Arc Diagrams noch ausgenutzt werden.

Ein Nachteil der Unscharfen Suche und der Distanzsuche soll nicht unerwähnt bleiben. Das im Abschnitt 4.2.2 vorgestellte Verfahren zur Beschleunigung der univariaten Mustersuche kann nicht zum Einsatz kommen, weil die vorgestellten Algorithmen der

Wahrscheinlichkeitsgestützten Suche und der Karp-Rabin Suche nicht sinnvoll eingesetzt werden können. Die Unscharfe- und die Distanzsuche sind somit vergleichsweise langsam. Für die Zukunft gilt es zu untersuchen, ob es möglich ist, die Wahrscheinlichkeitsgestützte Suche und den Karp-Rabin Algorithmus adäquat anzupassen.

4.2.4 Möglichkeiten zur Suche nach multivariaten Mustern

Bislang wurde sich der Verbesserung und der Erweiterung der univariaten Mustersuche durch Beschleunigung und Steigerung der Flexibilität gewidmet. Diese Analogie konnte auch problemlos für multivariate Zeitreihen genutzt werden, solange die Merkmalreihen nacheinander und getrennt betrachtet wurden.

Folgend sollen multivariate Muster gefunden werden. Die Suche nach multivariaten Mustern wurde in Abschnitt 4.2.1 als zweite Stufe der Mustersuche bezeichnet. Die Merkmalreihen müssen dazu im Zusammenhang betrachtet werden. Es ist erstrebenswert, eine Analogie zur Suche in Zeichenfolgen zu erhalten. Auf diese Weise könnten die bewährten String-Matching-Algorithmen, das vorgestellte Beschleunigungskonzept und die Flexible Suche weiterhin eingesetzt werden. Die Umstände einer solchen Erhaltung werden nun beschrieben.

Für die univariate Suche wurde eine Ausprägung der jeweils untersuchten Merkmalreihe pro Beobachtungspunkt betrachtet. Um einen Zusammenhang zwischen den Merkmalen herzustellen, wird ein Vektor aller untersuchten Merkmalreihen pro Beobachtungspunkt betrachtet. Angelehnt an das String-Matching-Problem (vgl. 4.2.2) wird dazu das Vektor-Matching-Problem verbal beschrieben:

> Es sind zwei zeitlich geordnete Mengen von Vektoren gegeben, die Vektormenge $\overrightarrow{vS} = \{\overrightarrow{vS_0}, ..., \overrightarrow{vS_{k-1}}\}$ und die Mustervektormenge $\overrightarrow{vX} = \{\overrightarrow{vX_0}, ..., \overrightarrow{vX_{p-1}}\}$. Im weiteren Verlauf der Arbeit sei die Anzahl *p* der Vektoren von $\overrightarrow{vX}$ als Länge von $\overrightarrow{vX}$ bezeichnet. Ein *Fenster* W_i ist eine Untermenge von Vektoren von $\overrightarrow{vS}$ der Länge *p*, die an Position *i* beginnt. Ein Fenster W_i, das mit $\overrightarrow{vX}$ übereinstimmt, heißt *Vorkommen* von $\overrightarrow{vX}$ an Position *i.* Gesucht sind alle Vorkommen der Mustervektormenge $\overrightarrow{vX}$ in der Vektormenge $\overrightarrow{vS}$. Diese Vorkommen sollen multivariate Muster genannt werden.

String-Matching-Algorithmen können nach dieser Definition für eine Suche nach multivariaten Mustern verwendet werden. Dadurch ist auch das vorgestellte Beschleunigungskonzept verwendbar (vgl. 4.2.2). Es ist jedoch zu beachten, dass pro Beobachtungspunkt nun nicht mehr eine Ausprägung, sondern alle Elemente des Vektors des Beobachtungspunktes zu vergleichen sind. Die Anzahl der Elemente jedes Vektors entspricht dabei der Anzahl der vorhandenen Merkmale der

multivariaten Zeitreihe. Abbildung 4-9 versucht eine Darstellung dieses Sachverhaltes. Eine multivariate Zeitreihe enthalte drei Merkmale. Es soll ein multivariates Muster der Länge eins gefunden werden. Die Mustervektormenge $\overrightarrow{vX}$ enthält also einen Vektor. In Abbildung 4-9 ist dargestellt, dass alle Elemente dieses Mustervektors mit den Elementen des Vektors im aktuellen Suchfenster W_2 verglichen werden. Ein multivariates Muster wird im dargestellten Fall nicht gefunden, da das dritte Element des Vektors nicht übereinstimmt.

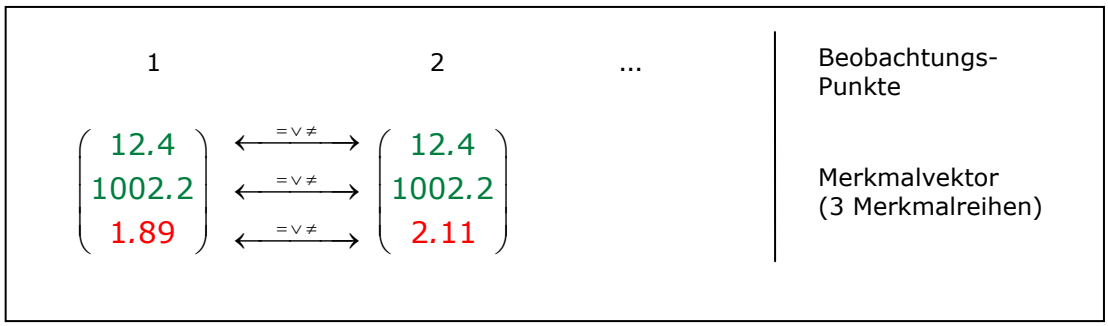

Abbildung 4-9: Die Suche multivariater Muster nach dem Vektor-Matching-Problem

In 4.2.3 wurde deutlich gemacht, dass der Umfang des Wertebereichs einen großen Einfluss auf das Finden von univariaten Musterpaaren hat. Dieser Einfluss ist für multivariate Muster noch gesteigert. Abbildung 4-9 zeigt z.B., dass für ein multivariates Muster der Länge eins bereits drei Ausprägungen übereinstimmen müssen. Daher wird die Integration der Suche nach Abschnitt 4.2.3 für die multivariate Mustersuche empfohlen. Die Unscharfe Suche wird dazu auf Vektoren und nicht mehr auf einzelnen Ausprägungen angewendet. Es wird vorgeschlagen, um jedes Element eines Vektors ein Unschärfeintervall zu legen. Das Vorgehen zur Festlegung dieses Unschärfeintervalls ist analog zu 4.2.3 durchführbar.

Auch die Distanzsuche soll zur Steigerung der Flexibilität der multivariaten Suche beitragen. Bislang liegt ein multivariates Muster nur vor, wenn alle Elemente der Vektoren betrachteter Beobachtungspunkte übereinstimmen. Dies ist ein Nachteil, weil z.B. ein multivariates Muster bezüglich zweier Merkmale in einer Zeitreihe mit vier Merkmalen bislang nicht erkannt werden kann. Um diesen Nachteil zu beheben, soll die *Dimensionalität eines multivariaten Musters* eingeführt werden. Die Dimensionalität eines multivariaten Musters sei die Anzahl übereinstimmender Elemente der Vektoren dieses Musters. Die Distanzsuche kann für die Suche bezüglich unterschiedlicher Dimensionalitäten eingesetzt werden. Dazu wird der normierte Übereinstimmungsfaktor $\ddot{u}_N$ der Distanzsuche (vgl. 4.2.3) an die gesuchte Dimensionalität angepasst. Sollen z.B. für die Abbildung 4-9 alle möglichen 2-dimensionalen multivariaten Muster gefunden werden, ist $\ddot{u}_N = {}^2/_3$ (1-[1Fehler/3Elemente]) zu wählen. Abbildung 4-9 zeigt grün

dargestellt ein gefundenes 2-dimensionales, multivariates Muster bezüglich der ersten beiden Merkmalreihen.

Im Verlauf dieses Abschnitts wurde eine eigene Definition für multivariate Muster geschaffen. Diese Definition ermöglicht es, bestehende String-Matching-Algorithmen weiterhin zu verwenden. Auch das erarbeitete Beschleunigungskonzept und die flexible Suche sind dadurch einsetzbar. Es ist außerdem möglich, Zusammenhänge zu finden, welche sich nicht über alle Merkmale der multivariaten Zeitreihen erstrecken. Dazu wurde die Dimensionalität multivariater Muster eingeführt.

Dennoch stoßen die bislang vorgestellten Verfahren für komplexe multivariate Zusammenhänge schnell an ihre Grenzen. Infolgedessen ist es nötig, leistungsstärkere Verfahren für eine Suche nach multivariaten Mustern einzusetzen. In der Literatur existiert eine Vielzahl verschiedener Methoden die, basierend auf unterschiedlichen Herangehensweisen zur Informationsanalyse, eingesetzt werden. Im Zusammenhang mit der Untersuchung großer Informationsmengen lassen sich dabei unter anderem nicht-visuelle Analyseverfahren, wie z.B. mathematisch-statistische Ansätze, Methoden der KI sowie weitere Techniken, die unter dem Begriff Data Mining zusammengefasst werden, anwenden. Ziel dieser Verfahren ist die datengesteuerte Entdeckung und Modellierung der in großen Datenvolumen verborgenen Muster und Zusammenhänge. Momentan zeichnet sich der Trend der Kombination bzw. Integration von nicht-visuellen Mining Methoden mit Visualisierungsmethoden als aktueller Forschungsschwerpunkt ab. Dieses Gebiet wird auch als Visuelles Data Mining bezeichnet. [An01] klassifiziert die Visuellen Data Mining Ansätze in drei Gruppen, abhängig davon ob:

1. Visualisierungstechniken bevor oder unabhängig von Data Mining Algorithmen angewendet werden,
2. Visualisierungstechniken nach Anwendung von Data Mining Algorithmen für die Darstellung der Ergebnisse angewendet werden,
3. Visualisierungstechniken während des Data Mining Prozesses angewendet werden, in der Data Mining- und Visualisierungsprozess eng verknüpft sind.

Ziel aller drei Gruppen ist es, Leistungspotential und Stärken der jeweiligen Richtungen zu kombinieren. Diesem Trend folgend, soll in dieser Arbeit eine Integration von mathematisch-statistischen Verfahren für die multivariate Analyse in die Arc Diagram Technik vorgeschlagen werden.

Analog zu Ankerst [An01] soll bezogen auf multivariate Muster die Arc Diagram Technik entweder zur Suche nach multivariaten Zusammenhängen (1. Gruppe) oder zur Darstellung dieser (2. Gruppe) verwendet werden. Das Ziel dabei ist es die Leistungsstärke der mathematischen Verfahren auszunutzen und die Schwäche der fehlenden

bildlichen Darstellung, mit Hilfe der Arc Diagram Technik zu kompensieren.

Es ist dazu nötig zunächst einen Überblick über die mathematisch-statistischen Verfahren aus der Gruppe der Data Mining Methoden zu geben. Auf eine Vorstellung der genannten Verfahren soll verzichtet werden. Für eine genaue Formulierung des mathematischen Modells der Verfahren sei z.B. auf [Ba06] verwiesen.

Backhaus et al. [Ba06] teilen die mathematisch-statistischen Methoden für die multivariate Analyse vor einem anwendungsorientierten Hintergrund in primär *strukturen-entdeckende* und in primär *strukturen-prüfende Verfahren* ein. Es sei erwähnt, dass eine überschneidungsfreie Zuordnung der Verfahren in der Praxis nicht immer möglich ist, da sich die Ziele dieser Verfahren zum Teil überlagern. Trotzdem soll diese Klassifikation übernommen und kurz vorgestellt werden, da sie sich für den Verlauf der weiteren Betrachtungen als günstig erweisen wird.

1. Strukturen-prüfende Verfahren haben das primäre Ziel, Zusammenhänge zwischen Merkmalen zu überprüfen. Dies setzt voraus, dass der Anwender a priori eine Vorstellung der kausalen Zusammenhänge zwischen den Merkmalen hat. Diese Vorstellung soll mit Hilfe geeigneter multivariater Verfahren überprüft werden. Als Vertreter dieser Verfahren seien die Regressionsanalyse, die Varianzanalyse und die Diskriminanzanalyse genannt.
2. Strukturen-entdeckende Verfahren haben das primäre Ziel, Zusammenhänge zwischen Merkmalen zu entdecken. Der Anwender besitzt a priori keine Vorstellung darüber, welche Beziehungszusammenhänge zwischen Merkmalen existieren. Daher muss er, anders als bei den strukturen-prüfenden Verfahren, die zu betrachtenden Merkmale auch nicht in abhängige und unabhängige einteilen. Um eine Verwechslung mit den bereits eingeführten abhängigen und unabhängigen Variablen im Abschnitt 2.2.1 auszuschließen, sollen die alternativen Bezeichnungen *erklärtes* und *erklärendes* Merkmal eingeführt werden[1]. Verfahren, die mögliche Zusammenhänge aufdecken können, sind zum Beispiel die Faktorenanalyse oder die Clusteranalyse.

Es soll nun eine Diskussion über die Nutzbarkeit und Integration solcher Verfahren für die Visualisierung multivariater Zeitreihen auf Basis der Arc Diagrams erfolgen. Die erste Gruppe der Ansätze nach [An01] stellt die Visualisierungstechnik an den Anfang des Visuellen Data Mining Prozesses. Für die vorliegende Arbeit heißt das, dass die Arc Diagram Darstellung vor den mathematisch-statistischen Methoden durchgeführt

[1] Die erklärenden Merkmale wirken auf die erklärten Merkmale ein und legen damit die Richtung eines kausalen Zusammenhanges fest.

werden soll. Die Arc Diagrams funktionieren bei diesem Ansatz als visuelles muster-entdeckendes Verfahren. Die nicht-visuellen mathematischen Methoden sind der Arc Diagram Darstellung nachgestellt. Es wird vorgeschlagen hier speziell die primär struktur-prüfenden Verfahren nach [Ba06] einzusetzen. Das Positive dieses Ansatzes ist, dass ein Nutzer sich mit Hilfe der bildlichen Darstellung durch Arc Diagrams eine a priori Vorstellung der multivariaten Zusammenhänge in den Merkmalreihen machen kann. Diese Zusammenhänge können dann durch die vorgestellten struktur-prüfenden Verfahren nachgewiesen oder widerlegt werden. Eine Darstellungsvariante, welche diesen Visuellen Data Mining Ansatz aufgreift, soll im Abschnitt 4.3.2 vorgestellt werden.

Für eine Integration mathematisch-statistischer Verfahren in die Arc Diagram Technik wird im Zuge dieser Arbeit der zweite Ansatz der Klassifikation nach [An01] favorisiert. Es wurde dargestellt, dass die Arc Diagram Technik in zwei Schritten abläuft. Der erste Schritt sucht alle essentiellen Paare. Der zweite Schritt stellt die gefundenen Paare mit Hilfe von Bögen dar.

Im ersten Schritt sollen neben den Analyseverfahren zur Suche nach univariaten Mustern künftig mathematisch-statistische Verfahren für die Suche nach multivariaten Zusammenhängen benutzt werden. Zur Suche nach multivariaten Mustern werden speziell die primär struktur-entdeckenden Verfahren nach [Ba06] empfohlen. Sie sind sehr leistungsstarke Methoden, welche auch komplexe multivariate Zusammenhänge in Merkmalen finden können.

Im zweiten Schritt werden gefundene uni- und multivariate Muster gemeinsam dargestellt. Dabei wird die schnelle visuelle Wahrnehmung und intuitive Verarbeitung von bildlichen Darstellungen des menschlichen Gehirns effektiv ausgenutzt (vgl. Abschnitt 2.1). Im Abschnitt 4.3.3 soll eine Darstellungstechnik auf Basis der Arc Diagrams vorgestellt werden, welche den favorisierten zweiten Visuellen Data Mining Ansatz umsetzt.

Zusammenfassung und Fazit

Dieser Abschnitt hatte die Aufgabe, die Möglichkeiten der multivariaten Suche nach Mustern zu analysieren. Dazu wurde, auf Basis einer eigenen Definition, die Suche multivariater Muster diskutiert. Diese Suche ist einfach und wird für die Implementierungen dieser Arbeit auch verwendet. Dennoch wurde deutlich gemacht, dass diese Suche für komplexe, multivariate Zusammenhänge weniger geeignet ist. Um diese Zusammenhänge zu enthüllen, wurde die Integration von Data Mining Verfahren für die multivariate Analyse in die Arc Diagrams vorgeschlagen. Zwei verschiedene Ansätze angelehnt an Ankerst wurden diskutiert, um diese Integration zu vollziehen. Im ersten Ansatz sollen Arc Diagrams primär multivariate Muster entdecken, die Mining Methoden diese prüfen. Im zweiten Ansatz ist dieser Sachverhalt genau umgekehrt. Für beide Ansätze sollen in den Abschnitten 4.3.2 und 4.3.3 Darstellungsvarianten diskutiert werden.

4.3 Lösungen zur Darstellung von multivariaten Zeitreihen

Im Abschnitt 4.2 wurde die Arc Diagram Technik als zweistufiges Verfahren beschrieben. Die erste Stufe bildet die Mustersuche. Die Möglichkeiten der univariaten und der multivariaten Suche für Arc Diagrams wurden erarbeitet. In diesem Abschnitt wird die zweite Stufe im Mittelpunkt stehen - Die Darstellung der gefundenen Muster. Im Abschnitt 4.1 wurde die grundlegende Anforderung an eine Darstellung formuliert: Die Darstellung uni- und multivariater Muster soll effektiv auf dem begrenzten Platz eines Ausgabegerätes erfolgen. Nach [Sp06] sind häufig mehr Informationen darzustellen, als auf einem Ausgabegerät dafür Platz zur Verfügung steht. Statt einer einzelnen Reihe sollen multivariate Reihen mit Arc Diagrams dargestellt werden. Das „presentation problem" nach [Sp06] ist deshalb noch verschärft. Aus diesem Grund werden im Abschnitt 4.3.1 allgemeine Verbesserungen der Darstellung erörtert. Die Abschnitte 4.3.2 und 4.3.3 diskutieren dann die zuvor erwähnten Darstellungsvarianten der Arc Diagrams als visuelle Data Mining Methode.

4.3.1 Allgemeine Verbesserungen der Arc Diagrams

Dieser Abschnitt diskutiert grundlegende Verbesserungen der Arc Diagrams. Diese Verbesserungen sind speziell auf die Darstellung von multivariaten Zeitreihen ausgerichtet. Nachfolgende Abschnitte sollen auf diesem Abschnitt aufbauen. Es werden drei Möglichkeiten einer Verbesserung erörtert:

- Die Reduktion der dargestellten Mustermenge,
- die Variation der Anordnung der Arc Diagrams und
- die Verschlüsselung zusätzlicher Datenwerte und Parameter.

Im Abschnitt 3.4 wurde erwähnt, dass die Arc Diagrams eine vollständige Darstellungstechnik sind. Allerdings wird nur eine Teilmenge aller möglichen Muster dargestellt. Für multivariate Zeitreihen müssen weitere Kompromisse bei der Darstellung eingegangen werden, um eine Darstellung mit mehr als einer einzigen Merkmalreihe nicht zu überladen. Die Teilmenge der angezeigten Musterpaare soll weiter eingeschränkt werden. Dafür wird vorgeschlagen, Wiederholungsbereiche R für eine Suche und eine Darstellung in multivariaten Zeitreihen nicht zu berücksichtigen (vgl. Abschnitt 3.3).

[Wa02] führt Wiederholungsbereiche für den Fall von mehrfach direkt aufeinander folgenden Mustern ein. Dort sollen sie die Struktur einer Zeichenfolge besser beschreiben, als ein andernfalls entstehendes globales Muster. Abbildung 4-10 verdeutlicht diesen Zusammenhang. Die linke Abbildung zeigt einen Wiederholungsbereich als essentiell passendes Paar. Die rechte Abbildung zeigt ein maximal passendes als essentielles Paar. Für die Darstellung z.B. der Struktur binärer Folgen ist

die linke Darstellung intuitiver. Dort ist der Einsatz von Wiederholungsbereichen sinnvoll.

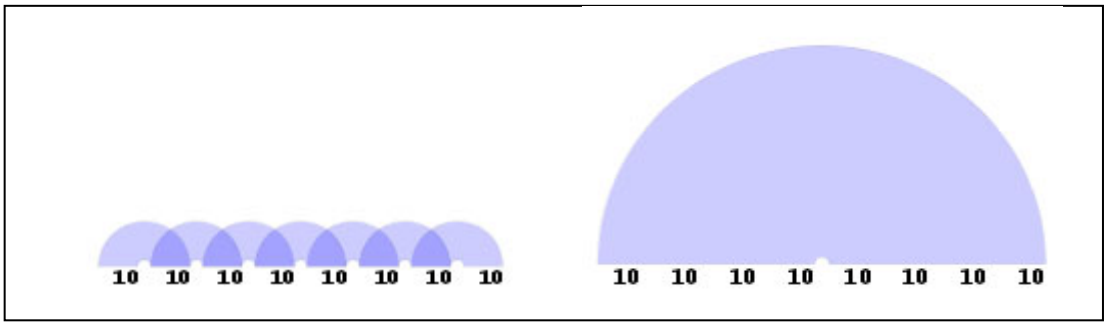

Abbildung 4-10: Wiederholungsbereich (links) vs. Maximal passendes Paar (rechts)

Die vorliegenden Zeitreihen sind nicht binär codiert. Sie haben einen wesentlich größeren Umfang des Wertebereichs als z.B. binäre Folgen. Die Wahrscheinlichkeit, dass gleiche Muster mehrfach unmittelbar aufeinander folgen ist wesentlich geringer. Falls solche unmittelbaren Wiederholungen dennoch vorhanden sind, sind sie in der Regel sehr kurz und für das Verständnis der Struktur einer Zeitreihe nicht wesentlich. Sie werden hinreichend durch ein maximal passendes Paar angezeigt (vgl. Abbildung 4-10).

Die ausbleibende Berücksichtigung von Wiederholungsbereichen bei der Suche und Darstellung stellt offensichtlich einen Kompromiss dar. Ein Nachteil dieses Kompromisses ist die vereinfachte Darstellung von Wiederholungsbereichen eines Musters durch maximal passende Paare. Demgegenüber stehen viele Vorteile: Ein wichtiger Vorteil ist eine weitere Beschleunigung der Mustersuche. Es muss nicht mehr getestet werden, ob maximal passende Paare in einem Wiederholungsbereich enthalten sind oder nicht (vgl. Abschnitt 3.3). Die Möglichkeit der Identifizierung von Mustern ist erhöht. Das maximal passende Paar kann von einem Nutzer besser erkannt werden, als die alternativen Wiederholungsbereiche. Dies gilt gerade auch für skalierte (verkleinerte) Darstellungen. Zusätzlich wird eine Darstellung durch ein einzelnes Muster nicht so schnell überladen, wie durch viele „kleinere" Muster.

Trotz der vorgestellten Reduzierung der dargestellten Mustermenge stoßen vollständige Darstellungen für große Datenmengen schnell an ihre Grenzen. Hier bieten sich unvollständige Darstellungen an. Diese repräsentieren eine echte Teilmenge der Daten. Durch die Visualisierung von Teilmengen der Daten treten Informationsverluste auf. Nach [SM00] können diese Informationsverluste z.B. durch das Bereitstellen von geeigneten Interaktionstechniken ausgeglichen werden. Interaktionstechniken stehen in Arc Diagrams bislang nicht zur Verfügung. Eine Reduktion der dargestellten Datenmenge durch unvollständige Darstellungen soll deshalb erst im Zuge der Integration von Interaktionstechniken im Abschnitt 4.4 diskutiert werden.

Bereits mehrfach in dieser Arbeit wurden Beispiele für die Darstellung einer einzelnen Merkmalreihe durch Arc Diagrams gezeigt (vgl. z.B. Abbildung 4-8). In der vorliegenden Arbeit ist eine multivariate Zeitreihe mit zehn Merkmalreihen mit Hilfe der Arc Diagrams darzustellen. Eine Möglichkeit ist, die Merkmalreihen mit Hilfe vorgestellter mathematisch-statistischer Methoden z.B. der Clusteranalyse zusammenzufassen und ein Arc Diagram, charakteristisch für alle Merkmalreihen darzustellen. Es ist aber auch möglich, für jede der Merkmalreihen ein Arc Diagram darzustellen und diese in geeigneter Weise anzuordnen.

In diesem Abschnitt wird der Fall betrachtet, für jede Merkmalreihe ein Arc Diagram darzustellen. Da Interaktionstechniken bislang nicht eingeführt wurden, sollen die Arc Diagrams zunächst automatisch auf die zur Verfügung stehende Darstellungsfläche eingepasst (skaliert) werden. Dehnungs- oder Stauchungsoperationen in der Darstellung seien nicht zulässig. Eine geeignete Anordnung der Arc Diagrams soll den Platz des Ausgabegerätes möglichst optimal ausnutzen. Dabei soll der formulierten Minimalanforderung einer Darstellung im Abschnitt 4.1, mindestens einen Überblick der gefundenen Muster zu zeigen, Sorge getragen werden.

[Tu83] beschreibt ein quantitatives Maß für Darstellungen – das *Daten-Tinte-Verhältnis*. Es ist definiert als das Verhältnis von abgebildeter Datenmenge zur verwendeten Menge an Druckerschwärze. Angelehnt an [Tu83] soll in dieser Arbeit das *Darstellungs-Pixel-Verhältnis* betrachtet werden. Es sei das Verhältnis verwendeter Pixel für die Darstellung zur vorhandenen Pixelmenge auf einem Ausgabegerät. Abbildung 4-11 zeigt dazu zwei mögliche Anordnungen. Das linke Bild zeigt exemplarisch vier Arc Diagrams neben- und untereinander angeordnet. Dies ist besonders für die Bildschirmauflösung von 16:9 eine akzeptable Lösung. Das Darstellungs-Pixel-Verhältnis ist höher als z.B. bei einer Anordnung der Arc Diagrams ausschließlich nebeneinander.

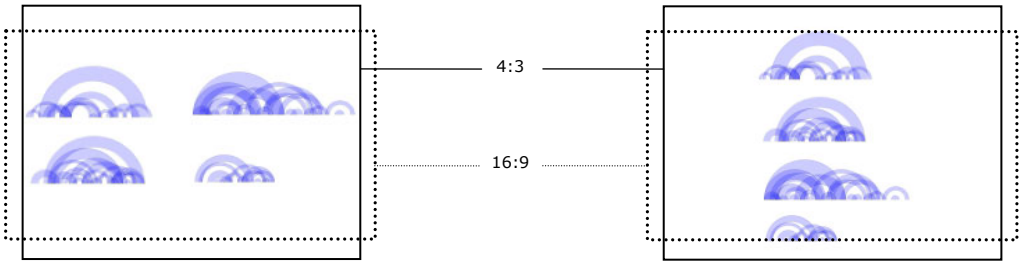

Abbildung 4-11: Zwei Variationen der Anordnung von vier Arc Diagrams

Das rechte Bild in Abbildung 4-11 ordnet die vier Arc Diagrams nur untereinander an. Diese Anordnung bietet sich z.B. für eine Bildschirmauflösung von 4:3 an. Die „linksbündige" Anordnung untereinander ist darüber hinaus gut für einen Vergleich der verschiedenen Merkmalreihenlängen oder für den Vergleich der Abstände der Musterpaare geeignet. Natürlich lassen sich viele weitere Möglichkeiten der Anordnung unterschiedlicher Anzahlen von Arc Diagrams finden. Dieses Problem ist eng verwandt mit dem Problem der

Platzierung einer Beschriftung (Labeling) in der Informationsdarstellung und dem Beschriftungsproblem der Kartographie. Das Finden einer optimalen Anordnung der Beschriftung ist dort, genauso wie hier, ein hoch komplexer Prozess, indem es viele Einflussfaktoren zu berücksichtigen gilt. Die angerissene Diskussion kann deshalb nicht erschöpfend sein. Auch die zwei vorgestellten Anordnungen in Abbildung 4-11 sind keinesfalls optimal. Sie sollen aber anregen, über weitere geeignete Anordnung nachzudenken.

Nachfolgend wird untersucht, auf welche Weise zusätzliche Datenwerte und Parameter in Arc Diagrams verschlüsselt werden können. Um eine Interpretation der Darstellung nicht wesentlich zu erschweren, soll darauf geachtet werden, dass entstehende Darstellungen die menschliche Wahrnehmung bestmöglich unterstützen.

In 3.3 und 3.4 wurde dargestellt, dass Arc Diagrams eine 2-dimensionale Technik sind, in der die Länge eines Musterpaares durch die Breite des entstehenden Bogens abgebildet wird. Die Höhe dieses Bogens ist abhängig vom Abstand der Musterpaare. Transparenz wurde eingesetzt, um trotz Überdeckungen und Überschneidungen von Mustern eine Zuordnung der Bögen zu den Musterpaaren zu ermöglichen.

Statt die Anordnung der Arc Diagrams für die zehn Merkmalreihen, wie zuvor beschrieben, auf zwei Dimensionen zu beschränken, könnte man eine dritte räumliche Dimension einsetzen. Nach [SM00] besteht wenig Einigkeit darüber, wann 3-dimensionale und wann 2-dimensionale Techniken im Vorteil sind. Dies soll problembezogen entschieden werden. Durch eine dritte Dimension erhält man die Möglichkeit, zusätzliche Datenwerte zu verschlüsseln. Dies stellt für die multivariate Datenmenge einen wichtigen Vorteil dar. Auch lebt der Mensch in einer 3-dimensionalen Umwelt. Er ist daran gewöhnt, Dinge 3-dimensional zu betrachten. Die Interpretation der 3-dimensionalen Darstellung wird also kein Hindernis darstellen. Trotzdem soll in dieser Arbeit eine 3-dimensionale Anordnung von Arc Diagrams nur bedingt befürwortet werden. Ein wichtiger Vorteil der Arc Diagrams ist seine Einfachheit der Darstellung von strukturellen Zusammenhängen. Durch die 2-dimensionale Anordnung entlang einer Achse kann ein Nutzer effektiv auf den „ersten Blick" Muster in Reihen erkennen. In 3-dimensionalen Darstellungen ist eine Interpretation auf den „ersten Blick" nicht immer möglich. Häufig kommt es z.B. zu Verdeckungen, welche durch Interaktions- und Navigationstechniken ausgeglichen werden müssen. Diese Interaktions- und Navigationstechniken sind gerade für 3-dimensionale Darstellung an hohe Anforderungen an Hard- und Software gekoppelt. Es sind Projektionen, Culling Operationen und Sichtbarkeitsberechnungen erforderlich. Bedingt durch die große multivariate Datenmenge der betrachteten Zeitreihen und den hohen Rechenaufwand der angesprochenen Operationen wird eine ausreichende Interaktionsfähigkeit nicht für jede eingesetzte Hard- und Software gewährleistet sein. In der vorliegenden Arbeit wird daher eine 3-dimensionale Darstellung der Arc Diagrams nicht berücksichtigt. Die Einfachheit und Möglichkeit der weiten Verbreitung durch die geringeren

Hardwareanforderungen soll über die Möglichkeit der Verschlüsselung zusätzlicher Parameter gestellt werden. Kann die Bereitstellung ausreichender Hard- und Softwarekapazitäten zugesichert werden, sind 3-dimensionale Darstellungen von Arc Diagrams aber eine geeignete Alternative.

Trotz der Beschränkung auf zwei Dimensionen soll es möglich sein, zusätzliche Informationen und Parameter in die Arc Diagrams zu integrieren. Dazu beschreibt [Be81] für statische 2-dimensionale Darstellungen prinzipiell acht verschieden visuelle Variablen eines graphischen Bildes:

- Die Position auf der Ebene,
- die Größe,
- der Helligkeitswert,
- die Musterung oder Textur,
- die Farbe,
- die Richtung oder Orientierung und
- die Form der Elemente.

All diese Variablen haben eine spezifische Wirkung bezüglich einer Darstellung. Beispielgebend sollen nur einige wenige Möglichkeiten der Variation ausgewählter Variablen, für die Darstellung von multivariaten Zeitreihen, genauer untersucht werden.

Bislang wurde für alle Bögen der Arc Diagrams dieselbe Farbe verwendet - in den meisten Abbildungen dieser Arbeit die Farbe „blau". Es ist sinnvoll über den Einsatz mehrerer Farben nachzudenken, um zusätzliche Parameter der multivariaten Zeitreihen zu verschlüsseln. Es wird vorgeschlagen die Farbe eines Bogens zu variieren, wenn er eine besondere Ausprägung einer Merkmalreihe enthält. Im Bereich der Zeitreihen sind unter anderem Maxima, Minima oder kritische Werte interessant. Nach [SM00] wird im Ingenieurbereich rot als Signalfarbe für Gefahr verwendet. So kann man z.B. rot einsetzen, um Musterpaare mit maximalen Ausprägungen einer Merkmalreihe zu kennzeichnen (vgl. Abbildung 4-12).

Zur Verschlüsselung von Informationen könnte man die Form der Elemente genauso wie die Farbe variieren. Die Bogenform der graphischen Primitive stellt aber einen fundamentalen Bestandteil der Arc Diagrams dar. Zum Beispiel ist die Höhe der Bögen direkt proportional zum Abstand der Musterpaare. Die Bogenform ist für ein Verständnis struktureller Zusammenhänge wesentlich. Für die Arc Diagrams soll eine Variation der Form deshalb nicht empfohlen werden.

Bislang ist die Transparenz für alle Bögen der Arc Diagrams gleichgroß festgelegt. Abschließend soll der Einsatz eines variablen Transparenzfaktors erwogen werden. Mit Hilfe einer variablen Transparenz können im Umfeld der Zeitreihen z.B. zusätzlich

Wichtigkeitsinformationen kodiert werden. Es ist vorstellbar, dass ein Nutzer speziell Muster einer bestimmten Länge sucht. Diese werden dann weniger stark transparent dargestellt, als die übrigen Muster. Die Wahl eines geeigneten Transparenzfaktors ist dabei sehr wichtig. Wird dieser Faktor zu niedrig gewählt, ist der Bogen fast vollständig lichtdurchlässig. Er ist dann nahezu unsichtbar. Ein zu hoher Transparenzfaktor eines Bogens sorgt dafür, dass alle Bögen hinter diesem fast vollständig verdeckt werden. Eine Zuordnung und Identifizierung der Bögen ist dann wesentlich erschwert.

Abbildung 4-12 zeigt noch eine andere interessante Möglichkeit des Einsatzes von Transparenz. Der Transparenzfaktor wird eingesetzt, um die Unschärfe eines Musters (vgl. 4.2.3) abzubilden. Ein exaktes Muster ist in der Abbildung mit einem Transparenzfaktor von 0.4 belegt. Mit zunehmender Unschärfe des Musters zum exakten Muster nimmt auch die Transparenz zu. Ein Muster, das gerade noch im festgelegten Unschärfeintervall liegt, soll einen Transparenzfaktor von 0.1 haben. Zusätzlich ist in Abbildung 4-12 eine Möglichkeit der Farbvariation dargestellt. Für Musterpaare, welche eine maximale Ausprägung der dargestellten Merkmalreihe enthalten, sind die Bögen rot anstatt blau eingefärbt.

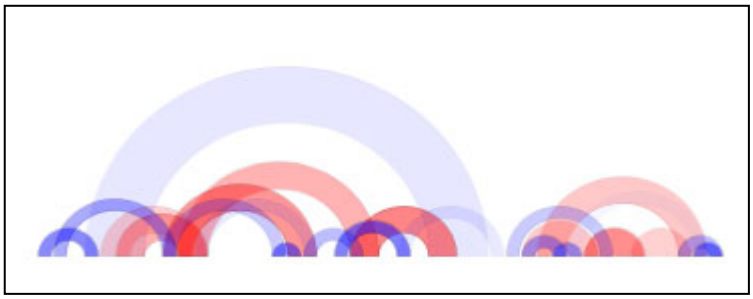

Abbildung 4-12: Arc Diagrams mit Variation der Transparenz und Farbe zur Verschlüsselung zusätzlicher Parameter

Zusammenfassung und Fazit

In diesem Abschnitt wurden grundsätzliche Verbesserungen der Arc Diagrams vorgeschlagen. Einige Verbesserungen sind speziell auf Zeitreihen zugeschnitten und auch nur für diese sinnvoll. So zum Beispiel die Nicht-Berücksichtigung von Wiederholungsbereichen. Andere Verbesserungen wurden vor dem Hintergrund von Zeitreihen entwickelt, sind aber durchaus allgemein nützlich. Hier sei die Farbvariation oder der Einsatz einer variablen Transparenz genannt. Ein Diskussionspunkt war speziell auf die Eigenschaft „multivariat" einer Zeitreihe ausgerichtet. Die Möglichkeit der unterschiedlichen Anordnung von Arc Diagrams. Dabei wurde sich bislang auf die Darstellung von univariaten Mustern durch Arc Diagrams beschränkt. Aufbauend auf den gesammelten Erkenntnissen, sollen nun zusätzlich auch multivariate Muster dargestellt werden.

4.3.2 Die Überlagerungsdarstellung für Arc Diagrams

Zur Suche nach multivariaten Mustern wird in dieser Arbeit eine eigene Definition multivariater Muster verwendet. Zeitgleich wird aber die Integration mathematisch-statistischer Methoden der Mustersuche in Arc Diagrams vorgeschlagen (vgl. 4.2.4). Dahingehend wurden zwei unterschiedliche Ansätze dieser Integration diskutiert. In diesem Abschnitt soll die Überlagerungsdarstellung vorgestellt werden. Sie ist in die erste Gruppe der Visuellen Data Mining Methoden nach [An01] bezüglich der multivariaten Mustersuche einsortierbar. Die Überlagerungsdarstellung soll wie folgt ablaufen:

1. Überlagerung der Arc Diagrams verschiedener Merkmalreihen zur multivariaten Strukturentdeckung.
2. Strukturprüfung durch mathematisch-statistische Verfahren.
3. Einsatz einer Variante der Arc Diagrams zur Darstellung multivariater Muster.

Im ersten Schritt soll ein Nutzer zunächst festlegen, zwischen welchen Merkmalreihen multivariate Muster gefunden werden sollen. Die Merkmalreihenanzahl soll sich zwischen minimal zwei und maximal allen vorhandenen Merkmalreihen der multivariaten Zeitreihe bewegen. Die Arc Diagrams der ausgewählten Merkmalreihen werden auf herkömmliche Weise ermittelt. Die ermittelten Arc Diagrams werden überlagert. Diese Überlagerung soll multivariate Zusammenhänge entdecken. Dafür erfolgt die Überlagerung linksbündig und in der Horizontalen so, dass sich genau die Ausprägungen der Reihen überlagern. Die Ausprägungen werden matrixförmig angeordnet unter den Bögen dargestellt werden. Abbildung 4-13 zeigt auf der linken Seite beispielhaft die Arc Diagrams zweier Merkmalreihen. Auf der rechten Seite der Abbildung ist die Überlagerung dieser Reihen auf diskutierte Weise dargestellt.

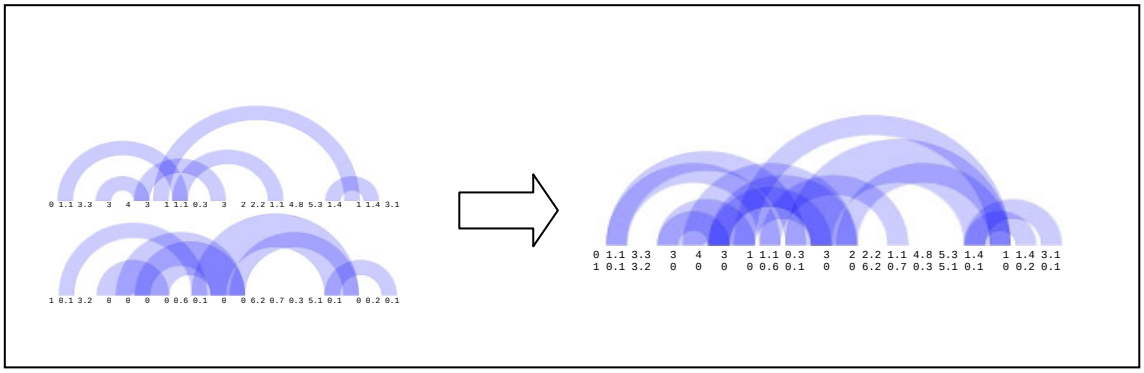

Abbildung 4-13: Prinzip der Überlagerung von zwei Arc Diagrams bei der Überlagerungsdarstellung

Aufgrund der Semitransparenz wirken Bereiche, in denen sich Bögen oder Teile von Bögen überlagern, dunkler. Weisen diese dunkleren Bereiche in etwa eine Bogenform auf, so ist es wahrscheinlich, dass an diesen Position multivariate Zusammenhänge auftreten. Anhand dieser dunkleren Bereiche sammelt ein Nutzer also Vermutungen über das mögliche Vorhandensein von multivariaten Mustern. Diese Vermutungen sind Grundlage des 2. Schrittes.

Im 2. Schritt soll der Einsatz von mathematisch-statistischen Verfahren erfolgen. Diese Verfahren sollen die gewonnenen Vermutungen untermauern oder widerlegen. Dazu ist in Abschnitt 4.2.4 insbesondere der Einsatz primär struktur-prüfender Verfahren empfohlen worden. Falls Vermutungen über multivariate Muster bestätigt werden können, werden diese im 3. Schritt dargestellt.

Grundlage des 3. Schrittes ist eine der Anforderungen an eine Darstellung aus 4.1: Uni- und multivariate Muster müssen voneinander unterscheidbar sein. Deshalb soll eine Variante der Arc Diagrams vorgeschlagen werden. Es werden zwei der visuellen Variablen nach [Be81] verändert (vgl. 4.3.1). Für multivariate Bögen soll eine andere Farbe, als für die univariaten Muster eingesetzt werden. Zusätzlich soll die Richtung der Bögen für multivariate Muster vertikal gespiegelt zu den Bögen der univariaten Muster verlaufen. Abbildung 4-14 zeigt ein mögliches Ergebnis der Überlagerungsdarstellung. Auf der linken Seite der Abbildung sind die Bögen von zwei Merkmalreihen überlagert. Die durch statistische Verfahren geprüften multivariaten Musterpaare sind grau, vertikal gespiegelt zu den univariaten blauen Musterpaaren, dargestellt. Die rechte Seite der Abbildung 4-14 zeigt einen Teilausschnitt der linken Abbildung vergrößert an. Es ist die matrixförmige Anordnung der Ausprägungen der Merkmalreihen erkennbar.

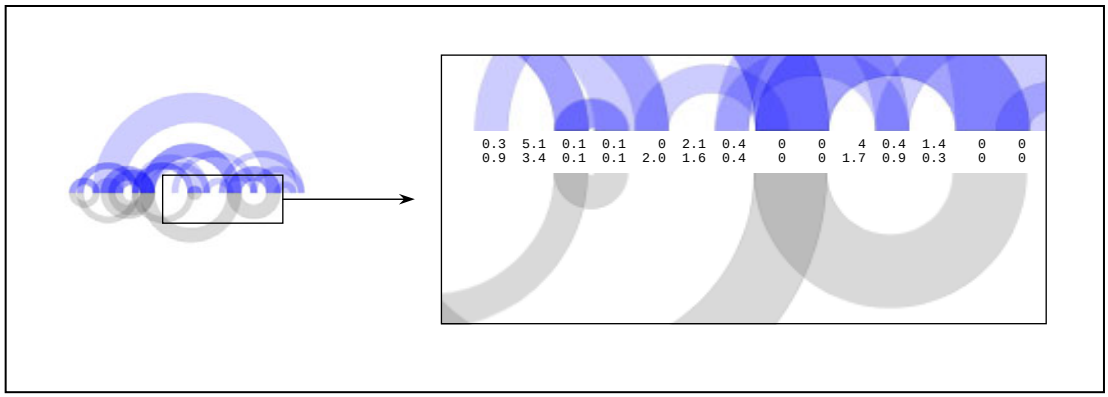

Abbildung 4-14: Überlagerungsdarstellung mit univariaten Mustern (blau) und multivariaten Mustern (grau)

An einem Beispiel soll eine interessante Variante der Überlagerungsdarstellung skizziert werden. Ziel ist die Ermittlung temporaler Zusammenhänge periodischer Schwankungen in multivariaten Zeitreihen. Die Musterpaare der Arc Diagrams sind für Zeitreihen als

periodische Schwankungen interpretierbar. Der Abstand der Musterpaare entspricht der Periodendauer. Ein temporaler Zusammenhang zwischen den Musterpaaren einzelner Merkmalreihen kann als multivariates Muster dieser Merkmale interpretiert werden. Für Zeitreihen ist die horizontale Verschiebung der Arc Diagrams entlang der X-Achse gleichzusetzen mit einer zeitlichen Verschiebung. Für eine horizontale Verschiebung ist es nötig die bisherige Überlagerungsdarstellung abzuändern: Die Restriktion der linksbündigen Überlagerung soll aufgehoben werden. Man hat nun die Möglichkeit der Verschiebung der einzelnen Arc Diagrams entlang der Zeitachse. Es soll solange horizontal verschoben werden, bis eine möglichst maximale Übereinstimmung der Bögen erreicht ist. Die maximale Übereinstimmung wird durch die visuelle Wahrnehmung eines Nutzers bestimmt. Im Prinzip entspricht dieses Vorgehen einer (visuellen) Kreuzkorrelation[2]. Im Fall der maximalen Übereinstimmung kann die Differenz aus erfolgter Verschiebung und linksbündiger Ausgangsdarstellung auf der X-Achse, als zeitliche Verschiebung der periodischen Schwankungen interpretiert werden.

Ein Beispiel soll den Nutzen der skizzierten Variante vergegenwärtigen. Gesucht ist die zeitliche Verschiebung der periodischen Schwankungen der Population zweier Tierarten in der Räuber-Beute-Beziehung. Räuber-Beute-Beziehungen sind ein Teilaspekt der Nahrungsnetze und Nahrungsketten in der Ökologie. Je mehr Beutetiere vorhanden sind, desto mehr Räuber finden Nahrung. Die Population der Räuber nimmt verschoben zur Population der Beutetiere zu. Gesucht ist die genaue Größenordnung dieser Verschiebung. Im Beispiel soll der Mäusebussard als Räuber dienen. Die Beute sei die Feldmaus. In Abbildung 4-15 ist auf der linken Seite eine denkbare, idealisierte, multivariate Zeitreihe von ermittelten Populationen der Bussarde und Mäuse in bestimmten Zeitabständen dargestellt. Für nicht idealisierte Zeitreihen wäre z.B. der Einsatz einer Flexiblen Suche (4.2.3) sinnvoll. Auf der rechten Seite sind zwei ableitbare Diagramme dieser Zeitreihe dargestellt. Bei den Arc Diagrams wurde sich auf die Darstellung von Musterpaaren der Länge eins beschränkt.

[2] Die Kreuzkorrelation ist ein Verfahren der Signalanalyse. Es wird dort z.B. zur Ermittlung von Laufzeitunterschieden zweier Signale eingesetzt. Für nähere Betrachtungen sei z.B. auf [OL05] verwiesen.

Zeit-punkt	Anzahl Bussarde	Anzahl Feldmäuse
1	*100*	*1000*
2	*150*	*800*
3	*100*	*600*
4	*70*	*800*
5	*100*	*1000*
6	*150*	*800*
7	*100*	*600*
8	*70*	*800*
9	*100*	*1000*

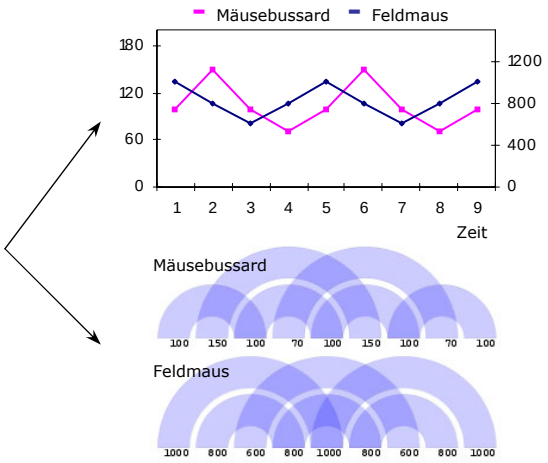

Abbildung 4-15: Darstellungsmöglichkeiten von Räuber-Beute-Beziehungen

Der vorgestellten Idee folgend, sollen jetzt die Arc Diagrams zunächst linksbündig überlagert werden. Anschließend soll eine Verschiebung auf der Zeitachse (X-Achse) erfolgen. Es soll solange verschoben werden, bis eine, visuell kontrollierte, maximale Übereinstimmung der einzelnen Bögen erreicht wurde. Zur Unterstützung des Nutzers bei der visuellen Kontrolle der Überlagerung während der Verschiebung werden graue Bögen dargestellt. Diese grauen Bögen werden genau dann gezeichnet, wenn Musterpaare betrachteter Merkmalreihen sich exakt überlagern. Abbildung 4-16 zeigt die linksbündige Ausgangsposition der Überlagerungsdarstellung (links). Auf der rechten Seite ist eine maximale Überlagerung dargestellt. Fünf graue Bögen weisen auf fünf exakte Übereinstimmungen hin. Die Merkmalreihe der Feldmäuse wurde für die maximale Übereinstimmung einen Schritt in Richtung der positiven Zeitachse verschoben. Dieser Verschiebungsschritt ist durch das Einfügen einer Ausprägung *V* in die Reihen kenntlich gemacht. Die Anzahl eines Verschiebungsschrittes drückt die Größenordnung der gegeneinander zeitlich verschobenen Musterpaare der Räuber-Beute-Beziehung aus. Die periodischen Schwankungen der Mäusebussarde folgen also den periodischen Schwankungen der Feldmäuse mit dem Abstand von 1 - z.B. Jahren oder Monaten, die Einheit der Zeitachse ist im Beispiel nicht näher bestimmt.

Es sei eingeräumt, dass die Verschiebung um eine Einheit zur Findung der periodischen Schwankung nicht besonders eindrucksvoll wirken mag. Durch genauere Betrachtung der Tabelle in Abbildung 4-5 ist es möglich die ermittelte Periodendifferenz sofort abzulesen. Auch möglich wäre es, die Linienzüge des Liniendiagramms zur Ermittlung der Periodendifferenz in Übereinstimmung zu bringen. Für größere, oft verrauschte Zeitreihen und höhere Differenzen der Periodendauer ist die vorgeschlagene Variante der Überlagerungsdarstellung aber im Vorteil gegenüber dem Versuch des Ablesens in Tabellen. So kann z.B. die flexible Suche unterstützend eingesetzt werden, um Fehler auszugleichen und

Toleranzen zuzulassen. Auch gegenüber der Verschiebung der Linienzüge ist die Überlagerungsdarstellung im Vorteil. Die Linienzüge in Liniendiagrammen müssen für eine effektive Überlagerung z.B. zunächst auf einer Wertbereichsachse (Y-Achse) zueinander geeignet skaliert werden. Für Arc Diagrams ist dies nicht erforderlich.

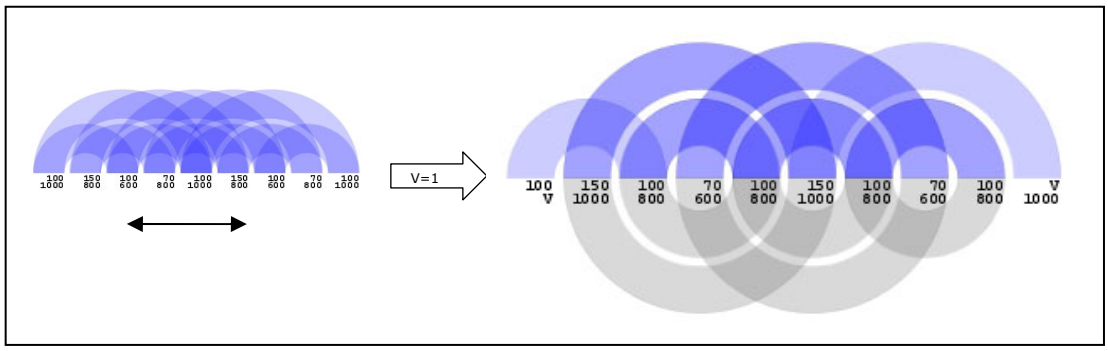

Abbildung 4-16: Darstellung einer linksbündigen Überlagerung (links) und einer optimalen Überlagerung durch Verschiebung (rechts)

Zusammenfassung und Fazit

Die Überlagerungsdarstellung nutzt die visuelle Fähigkeit des Menschen zur Erkennung von strukturellen Mustern. Multivariate Zusammenhänge werden erkannt, weil sie sich durch Helligkeitsunterschiede aus den überlagerten Arc Diagrams hervorheben. Dabei ist vorausgesetzt, dass alle Merkmalreihen gleich lang sind und jeweils genau eine Ausprägung pro Beobachtungspunkt besitzen. Diese Tatsache ist in vorliegender multivariater Zeitreihe gegeben. Andernfalls wird die Überlagerungsdarstellung nicht sinnvoll anwendbar sein.

Der Vorteil der Überlagerungsdarstellung ist seine „rasche" Durchführbarkeit. Der 1. Schritt ist effektiv durchführbar durch die menschliche Fähigkeit der intuitiven Interpretation bildhafter Darstellung. Mit Hilfe des 1. Schrittes ist im 2. Schritt eine gerichtete Suche nach multivariaten Mustern möglich. Es liegen Vermutungen über multivariate Muster vor. Die gezielte Überprüfung dieser Muster ist wesentlich schneller als die ungerichtete Suche nach ihnen. Nachteilig bei der Überlagerungsdarstellung ist die vergleichsweise „grobe" Suche nach multivariaten Mustern im 1. Schritt. Für die wachsende Anzahl an Merkmalen wird der Mensch zunehmend Probleme haben, Vermutung über multivariate Muster zu formulieren. Ein weiterer Nachteil ist die Tatsache, dass durch die Überlagerung, die Arc Diagrams der einzelnen Merkmalreihen nicht mehr visuell zu trennen sind. Dieser Nachteil soll aber in 4.4 behoben werden.

4.3.3 Die N-Eck Darstellung zur Erweiterung der Arc Diagrams

Neben der Überlagerungsdarstellung soll noch eine weitere Möglichkeit zur Visualisierung von Mustern in multivariaten Zeitreihen vorgestellt werden – Die N-Eck Darstellung.

Zunächst wird für jede der Merkmalreihen der multivariaten Zeitreihe ein Arc Diagram auf übliche Weise ermittelt. Anschließend wird ein regelmäßiges N-Eck[3] konstruiert, wobei jeder Seite genau ein Arc Diagram für eine Merkmalreihe zugeordnet ist. Damit wird das Äußere des N-Ecks zur Darstellung von univariaten Mustern durch Arc Diagrams benutzt. Das Innere des N-Ecks wird genutzt, um multivariate Zusammenhänge darzustellen. Dazu wird das Innere des N-Ecks, wie in Abbildung 4-17 dargestellt, unterteilt. Jedes gebildete Segment korrespondiert mit einer Merkmalreihe. Jede Ausprägung einer Merkmalreihe soll mit einem Sektor korrespondieren. Dafür wird jedes Segment auf die in Abbildung 4-17 dargestellte Weise in so viele Sektoren unterteilt, wie die korrespondiere Merkmalreihe Ausprägungen besitzen. Die Sektoren sollen der Verschlüsselung der Dimensionalität multivariater Muster dienen (vgl. 4.2.4). Zu diesem Zweck soll jeder Sektor zusätzlich von außen nach innen in Bereiche unterteilt werden. Die Einfärbung der Bereiche eines Sektors soll auftretende 2- bis n-dimensionale multivariate Muster repräsentieren. Ein Bereich wird genau dann eingefärbt, wenn die mit dem Sektor korrespondierende Ausprägung an einem entsprechend dimensionalen multivariaten Muster beteiligt ist.

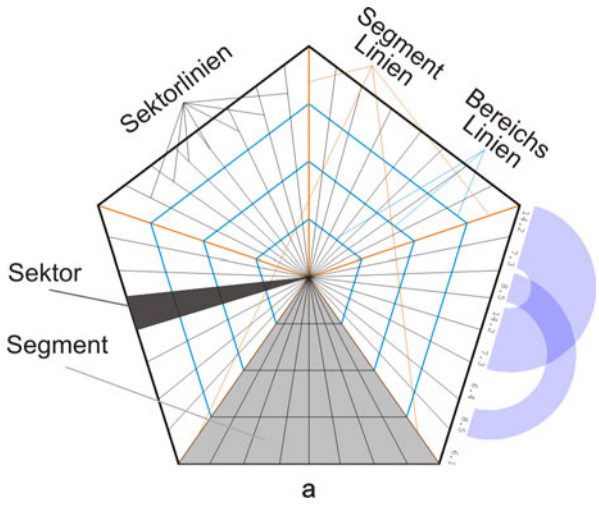

Abbildung 4-17: Ein regelmäßiges N-Eck

[3] In einem regelmäßigen N-Eck sind alle Kanten gleichlang und alle Innenwinkel gleich groß.

Die Funktion und der Nutzen möglicher N-Eck Darstellungen soll durch ein Beispiel vor Augen geführt werden. Es seien exemplarisch fünf Merkmalreihen der Temperatur zu visualisieren. Jede Merkmalreihe enthält acht Temperaturwerte. Jedes Segment enthält also acht Sektoren, jeder Sektor ist in vier Bereiche unterteilt. Die vier Bereiche jedes Sektors verschlüsseln 2- bis 5-dimensionale multivariate Muster.

Abbildung 4-18 zeigt die Arc Diagrams außen an den Kanten des N-Ecks angeordnet. Im Inneren des N-Ecks sind multivariate Muster durch farbige Bereiche dargestellt. Der äußerste Bereich jedes Sektors ist z.B. eingefärbt, wenn die korrespondierende Ausprägung an mindestens einem 2-dimensionalen multivariate Muster beteiligt ist. Die Dimensionalität der Muster nimmt nach innen gerichtet zu. Anstatt eine einzige Farbe zur Einfärbung der Bereiche einzusetzen, ist in Abbildung 4-18 zusätzlich eine Farbskalierung durchgeführt. Die Höhe der Temperatur der einzelnen an multivariaten Mustern beteiligten Ausprägungen ist auf eine Farbskala abgebildet. Dies erlaubt die Extraktion zusätzlicher Informationen aus der Darstellung, dass z.B. die höchste auftretende Temperatur von 30°C an einem 4-dimensionalen multivariaten Muster beteiligt ist.

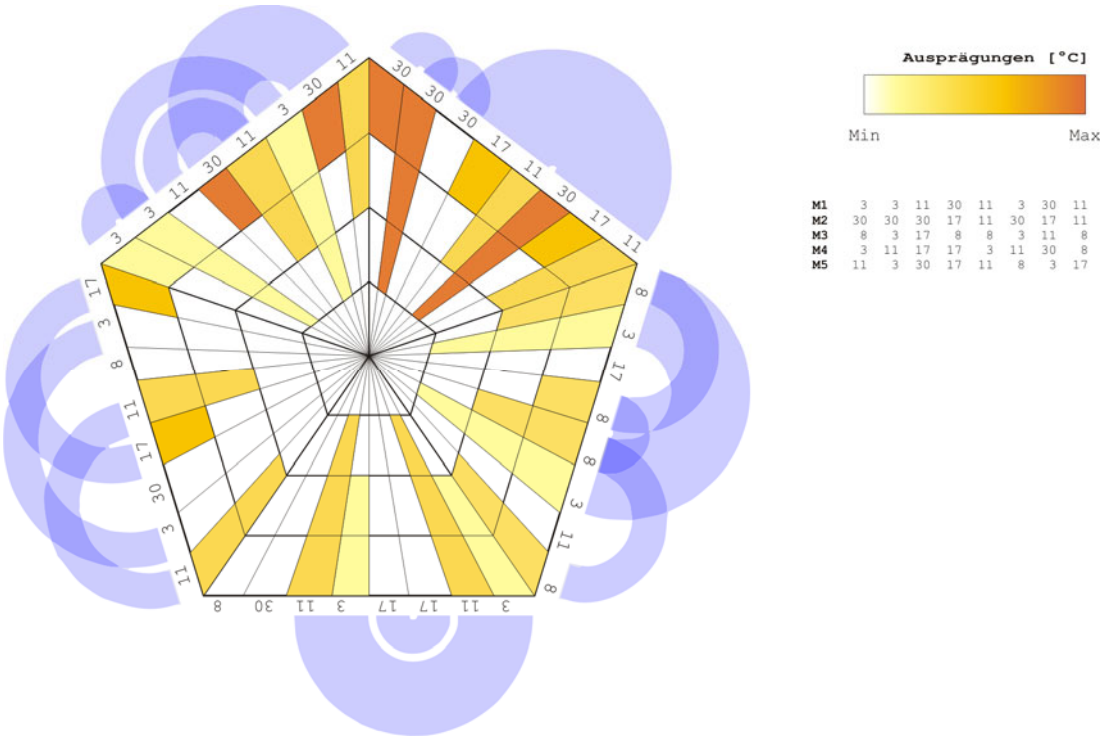

Abbildung 4-18: N-Eck Darstellung für fünf Merkmalreihen mit zusätzlicher Farbskalierung der multivariaten Ausprägungen

Zusammenfassung und Fazit

Die N-Eck Darstellung ist bezüglich der multivariaten Mustersuche in die zweite Gruppe der Visuellen Data Mining Methoden nach [An01]

einsortierbar. Das Finden von Mustern wird von nicht-visuellen Verfahren übernommen. Diese ungerichtete Suche ist vergleichsweise langsam, dafür aber sehr exakt. Anschließend wird die menschliche Fähigkeit der intuitiven Wahrnehmung von Bildern für die Darstellung ermittelter Muster genutzt. Dazu werden die univariate Muster außerhalb, die multivariaten Muster innerhalb eines N-Ecks dargestellt. Die farblich gekennzeichneten Bereiche innerhalb des N-Ecks ermöglichen eine Zuordnung von Ausprägungen zu multivariaten Mustern. Darüber hinaus ermöglichen sie die Verschlüsselung der Dimensionalität multivariater Muster und weiterer Parameter. Dies stellt einen wichtigen Vorteil dar.

Sowohl die Überlagerungs-, als auch die N-Eck Darstellung können bislang lediglich die genannte Minimalanforderung einer Darstellung aus 4.1 erfüllen. Sie stellen nur einen Überblick der ermittelten Zusammenhänge und Muster bereit. Je nach Merkmalreihenumfang und Länge muss eine Überlagerungs- oder N-Eck Darstellung stark verkleinert (skaliert) werden, um in ein Darstellungsfenster zu passen. 2.1.3 stellte dar, dass entweder die Auflösung des graphischen Anzeigegerätes oder des menschlichen Auges für starke Verkleinerungen versagen. Feine Details können nicht dargestellt oder erkannt werden. Zur erschöpfenden Exploration ist es deshalb nötig, Interaktionstechniken zu integrieren.

4.4 Integration von Interaktionstechniken

Bislang sind in die Arc Diagrams keine Techniken zur Interaktion integriert. Dieser Abschnitt zeigt die Notwendigkeit und diskutiert die Möglichkeiten einer Interaktionsintegration. Ausgehend von den Anforderungen an diese Integration in 4.1 soll es dem Nutzer möglich sein, sowohl in den Prozess der Mustersuche, als auch in den Prozess der Musterdarstellung zielführend einzugreifen. Dazu sollen die betrachteten Interaktionstechniken in

- Interaktionstechniken auf den Daten,
  - Musterlänge
  - Flexible Suche nach Mustern
  - interaktive Selektion von Datensätzen
- Navigationstechniken auf der Darstellung und die
  - Scrolling und Panning
  - Zoom
- Identifikationstechniken
  - Highlighting
  - Beschriftungstechniken

gruppiert werden.

Diese Gruppen sollen einzeln erläutert werden. Die Suche nach Musterpaaren in Arc Diagrams kann viele spezielle Ziele haben. Es ist z.B. möglich, dass einen Nutzer nur Musterpaare einer bestimmten Länge interessieren. Es ist auch möglich, dass ihn nur Muster in echten

Teilmengen der gesamten Datenmenge interessieren. Automatisch getroffene Voreinstellungen der Mustersuche in Arc Diagrams können deshalb nicht immer gut geeignet sein. Der Nutzer muss in die Parametrisierung der Mustersuche eingreifen können. Ebenso wichtig ist die Möglichkeit des Eingreifens in die Mustersuche rückgekoppelt mit einer Darstellung. Oft sieht ein Nutzer erst in der Darstellung, was ihn eigentlich interessiert. Oder er bemerkt, dass die Darstellung überladen ist und keine Informationsextraktion zulässt. Auch deshalb müssen Interaktionstechniken der Mustersuche auf der Datenbasis möglich sein.

Alle Interaktionstechniken bezüglich der Mustersuche sollen als Interaktionstechniken auf den Daten zusammengefasst werden. Dafür wurden in dieser Arbeit drei wichtige Möglichkeiten identifiziert. Der Nutzer muss die Länge der zu findenden Musterpaare interaktiv einstellen können. In 4.2.2 wurde dargestellt, dass für Muster minimale Längen von $p=1$ und die maximale Länge von $\lceil (k-1)/2 \rceil$ in Betracht kommt. Der Nutzer soll innerhalb dieses Intervalls einzelne Musterlängen oder Intervalle von Musterlängen für eine Mustersuche auswählen können.

Die zweite sinnvolle Möglichkeit der Interaktion auf den Daten ist die Parametrisierung der flexiblen Musterfindung. In 4.2.3 wurde die Notwendigkeit einer Flexiblen Suche nach Mustern dargestellt. Eine Interaktion auf dieser Flexiblen Suche ist nötig, um den Toleranzbereich der Suche nutzergesteuert festzulegen. Für die unscharfe Suche soll der Unschärfefaktor u interaktiv einstellbar sein, für die Distanzsuche der normierte Übereinstimmungsfaktor $\ddot{u}_N$ (vgl. 4.2.3).

Die letzte Interaktionsmöglichkeit auf den Daten soll dem Verlust von Informationen in der Darstellung entgegenwirken. Es wird die Möglichkeit einer interaktiven Selektierbarkeit von Datensätzen vorgeschlagen. Weil zwei unterschiedliche Möglichkeiten des Informationsverlustes betrachtet werden, soll zwischen der Selektion von Mengen der Daten oder der Muster unterschieden werden.

Ein erster Informationsverlust findet durch die möglicherweise zu starke Überlappung und Überschneidung von Bögen der Musterpaare in der Darstellung statt. Eine Selektion von Musterpaaren kann die Anzahl der dargestellten Muster verringern. Damit wird die Wahrscheinlichkeit einer Überlappung und Überschneidung vermindert.

Ein zweiter Informationsverlust findet durch die möglicherweise zu große Menge an dargestellten Merkmalreihen und Ausprägungen in Verbindung mit dem begrenzten Auflösungsvermögen des graphischen Ausgabegerätes und/oder des menschlichen Auges statt. Eine Selektion von Datensätzen der Datenmenge kann es ermöglichen, diese Informationsverluste zu verhindern. Soll nur eine Teilmenge der Datensätze angezeigt werden, können diese vergrößert skaliert in einem Fenster dargestellt werden. Die Anforderung an das Auflösungsvermögen des Ausgabegerätes und des Auges sinken.

Eine Selektion wird über die Spezifikation eines Filters festgelegt. Dieser stellt gewisse Bedingungen an die Menge der Daten oder Musterpaare. Abbildung 4-19 zeigt z.B. eine vollständige Musterpaarmenge (mit 685

Mustern) auf der linken Seite. Auf der rechten Seite ist eine Musterselektion dargestellt. Es werden nur genau die Musterpaare angezeigt deren Länge eins und deren Ausprägung „85" beträgt.

Es sei erwähnt, dass durch die Darstellung einer Teilmenge der Daten die Arc Diagrams zu einer unvollständigen Darstellungstechnik werden (vgl. 3.4). Der damit verbunden Informationsverlust kann aber direkt durch das Bereitstehen der vorgestellten Interaktionstechniken auf der Datenbasis ausgeglichen werden.

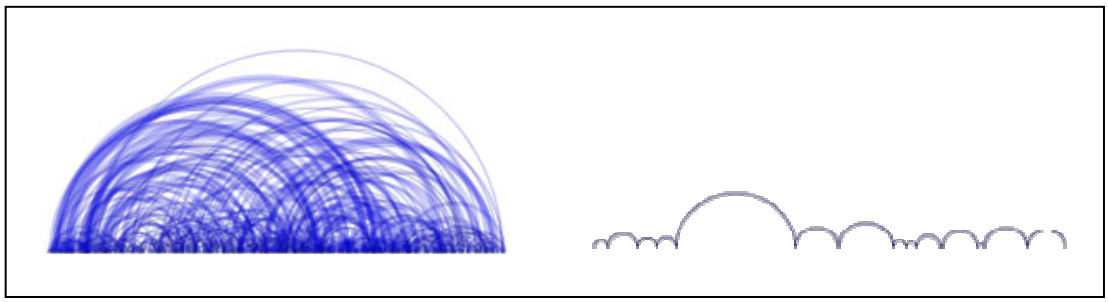

Abbildung 4-19: Arc Diagram mit vollständiger (links) und selektiver Musterauswahl (rechts)

In 2.1.3 wurde dargestellt, dass eine einzige Darstellung einer Menge von Daten nicht ausreicht. Es gilt viele Darstellungen und Teilsichten zu erzeugen, um eine vollständige Exploration der Daten zu erreichen. Um alle Darstellungen und Teilsichten anzeigen zu können, müssen dem Nutzer Interaktionstechniken auf der Darstellung bereitgestellt werden. Diese Interaktionstechniken sollen zur Gruppe der Navigationstechniken zusammengefasst werden.

Zur Navigation in Arc Diagrams multivariater Zeitreihen wurden in dieser Arbeit zwei wichtige Möglichkeiten identifiziert. Dem Nutzer muss die Gelegenheit des Zoomens und des Scrolling und Panning gegeben werden (vgl. 2.1.3). Durch eine Vergrößerung (Zoom) der Darstellung können auch feine Details angezeigt werden. Durch die Vergrößerung eines Teiles der Darstellung werden allerdings andere Teile des Bildes aus dem sichtbaren Ansichtsfenster des Ausgabegerätes verdrängt. Um diese Teile zu erreichen, soll der Einsatz des Scrolling und Panning vorgeschlagen werden.

2.1.3 nennt die mangelhafte Möglichkeit der Orientierung bezüglich der Gesamtdarstellung als Nachteil der Darstellung von Teilsichten. Dieser Nachteil soll behoben werden. Dazu wird vorgeschlagen, das „Übersicht und Detail" Konzept mit räumlicher Trennung (vgl. 2.1.3) zusammen mit dem Zoom und dem Scrolling und Panning umzusetzen. Der Detailbereich des Darstellungsfensters kann durch diese Techniken feine Details jeder vorhandenen Teilsicht anzeigen. Die Orientierung wird durch die Anzeige einer Gesamtdarstellung im Übersichtsbereich des Darstellungsfensters ermöglicht.

Für das „Übersicht und Detail" Konzept wird eine Semantische Skalierung vorgeschlagen (vgl. 2.1.3). Die geringste Auflösungsstufe soll im Übersichtsbereich angezeigt werden. Die feineren Auflösungsstufen in Abhängigkeit der aktuellen Vergrößerung im Detailbereich des Darstellungsfensters. 2.1.3 beschreibt eine Semantische Skalierung besonders dann als effektiv, wenn es gelingt die zu visualisierenden Daten geeignet zu gewichten. Mit der Länge eines Musterpaares nimmt in der Regel auch seine Bedeutung für das Erkennen der Gesamtstruktur einer Merkmalreihe zu. Eine geeignete Gewichtsfunktion ist also z.B. die Länge des Musters. In der Übersichtsdarstellung sollen nur Musterlängen p bis zu einer gewissen unteren Schranke angezeigt werden. Diese untere Schranke wird mit zunehmender Vergrößerung im Detailbereich bis zur unteren Schranke der Musterlänge $p=1$ abgesenkt. Für die vorliegenden multivariaten Zeitreihen wird in der Übersichtsdarstellung ein Schranke von $p=3$ zur Anzeige empfohlen. Aufgrund des Umfangs des Wertbereichs der Merkmalreihen kommen aber nur sehr wenige Musterpaar der Länge drei vor. Deshalb wird zusätzlich der Einsatz einer Flexibleren Suche (vgl. 4.2.3) für den Übersichtsbereich des Darstellungsfensters empfohlen. Die Wahrscheinlichkeit des Vorkommens von Musterpaaren mit einer Länge $p\geq3$ ist so erhöht. Zusätzlich sind die entstehenden „längeren" Muster durch die Flexiblere Suche optisch erfassbarer und somit für die Übersichtsdarstellung geeignet.

Abbildung 4-20 zeigt eine mögliche Umsetzung des „Übersicht und Detail" Konzepts mit Semantischer Skalierung. Im oberen Bereich des Bildschirms sind vier Arc Diagrams in einer Übersichtsdarstellung mit $p\geq3$ nebeneinander angeordnet. Der untere Teil des Bildschirms dient der Detaildarstellung. Es ist eine Teilsicht des Überblicksbereiches mit allen Musterpaaren sichtbar. Die aktuell angezeigte Teilsicht ist in der Übersichtsdarstellung zur Orientierung grau unterlegt.

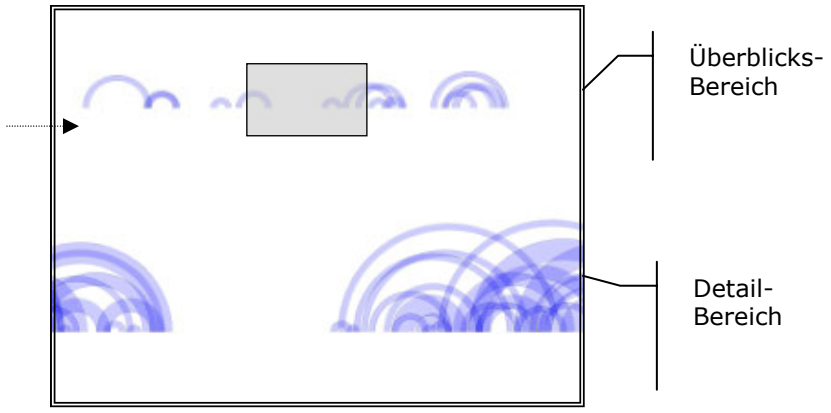

Abbildung 4-20: "Übersicht und Detail" Konzept zur Anzeige von Arc Diagrams

Es sei erwähnt, dass auch durch das Bilden von Teilsichten unvollständige Darstellungen entstehen. Aber auch hier kann der entstehende

Informationsverlust direkt durch die vorgestellten Navigationstechniken ausgeglichen werden.

In 4.3 ist die Tatsache des verschärften „presentation problems" für die Darstellung von multivariaten Zeitreihen dargestellt. Es sollen viele Merkmalreihen mit unterschiedlichen Ausprägungen auf einmal angezeigt werden. Die Zuordnung von Ausprägungen zu Bögen oder z.B. von Bögen zu unterschiedlichen Merkmalreihen ist nicht immer einfach. Es ist daher erforderlich interaktive Techniken bereitzustellen, welche eine klare Zuordnung von zusammengehörigen Darstellungselementen ermöglichen. Weiterhin soll es möglich sein, Zusatzinformationen über Bereiche von speziellem Interesse zu erhalten. Diese interaktiven Techniken sollen zur Gruppe der Identifikationstechniken zusammengefasst werden.

Für die vorgestellten Darstellungsvarianten in 4.3.2 und 4.3.3 sind spezielle Identifikationstechniken denkbar. Für die Überlagerungsdarstellung beispielsweise soll der Nutzer die Überlagerung der Arc Diagrams interaktiv von nicht bis vollständig überlagert steuern können. Für die N-Eck Darstellung ist z.B. eine interaktive Rotation sinnvoll, weil Musterpaare in der jeweils waagerechten Kante des N-Ecks leichter zu interpretieren und zu identifizieren sind.

Zwei Interaktionstechniken zur Identifikation sollen vorgestellt werden, welche allgemein für Arc Diagram Darstellungen interessant sind. Das Highlighting soll interaktiv dargestellte Elemente hervorheben, wenn sich der Mauszeiger darüber befindet. Denkbar ist die interaktive Einfärbung von Bögen oder z.B. die Hervorhebung der Ausprägung des Bogens.

Eine zweite sinnvolle Identifizierungstechnik für Arc Diagrams ist die Integration einer Beschriftungstechnik. Statische Beschriftungstechniken erzeugen möglichst optimal angeordnete, nicht überlappende Anordnungen von Labels. Um der Dynamik einer Darstellung durch Interaktionen eines Nutzers gerecht zu werden, sind statische Techniken zu rechenzeitintensiv. In dieser Arbeit wird deshalb die Integration von dynamischen Beschriftungstechniken vorgeschlagen. Sie stellen einen Kompromiss aus Ressourcenbedarf und möglichst geeigneter Anordnung der Labels dar. Als bekanntestes Beispiel sei der klassische „Infotip" genannt. Ein Label des Bogens soll an einer geeigneten Position erscheinen, wenn sich der Mauszeiger über einem Bogen befindet und verschwindet andernfalls wieder.

Zusammenfassung und Fazit

Die Möglichkeiten der Interaktion mit der Arc Diagram Technik stellt erhöhte Anforderungen an die verwendete Hard- und Software. Demgegenüber stehen aber viele Vorteile. Die Integration von Interaktionstechniken auf den Daten und von Navigationstechniken auf der Darstellung ermöglicht es, zielführend in den Prozess der Mustersuche und der Musterdarstellung einzugreifen. Die in 4.1 formulierte dritte Anforderung an die Arc Diagrams wird somit erreicht.

Darüber hinaus wird die Integration von Techniken zur Identifikation in die Arc Diagrams vorgeschlagen. Unter anderem aufgrund der hohen Informationsdichte durch die Darstellung vieler Arc Diagrams gleichzeitig sind sie ein wichtiges Hilfsmittel für einen Nutzer.

Kapitel 5

Implementierung

In diesem Kapitel sollen die Teilaspekte des vorgestellten Konzepts aus Kapitel vier in einem interaktiven Prototyp zusammengefasst werden. Im folgenden Abschnitt sind die Anforderungen an die Applikation zusammengefasst. Darauf aufbauend werden Architektur und Umsetzung der Applikation beschrieben. Abschließend soll in 5.3 eine kurze Anwendungsbeschreibung sowie in 5.4 ein Ausblick über mögliche zukünftige Implementierungsarbeiten gegeben werden.

5.1 Anforderungen

Oberstes Ziel ist, das Gesamtkonzept des vierten Kapitels widerzuspiegeln und seine Realisierbarkeit aufzuzeigen. Aus diesem Ziel lassen sich bestimmte Anforderungen ableiten. Eine Anforderung ist es, wichtige Komponenten der beschriebenen Funktionen zur Mustersuche und zur Musterdarstellung zu integrieren. Des Weiteren sollen beispielgebend mehrere Möglichkeiten der diskutierten Interaktionstechniken in die Arc Diagrams integriert werden. Im Rahmen der Umsetzung des Konzepts ist darauf zu achten, dass für eine Referenzhardware bei der grafischen Ausgabe und der Interaktion keine oder nur geringe Wartezeiten für den Nutzer entstehen. Andernfalls wäre die Forderung der Angemessenheit einer Darstellung verletzt (vgl. 2.1.1). Als Referenzhardware für die Implementierung dient ein PC mit einem Intel Pentium IV Prozessor getaktet mit 2.8 GHz. Der Rechner verfügt über einen Arbeitsspeicher von 512 MB und eine Grafikkarte mit einem Videospeicher von 64 MB. Weiterhin soll die Applikation einer großen Gruppe von Nutzern zugänglich sein. Daraus leiten sich zwei weitere Anforderungen an die Applikation ab: es muss möglichst plattformunabhängig ausführbar sein, und es muss intuitiv bedient werden können. Eine lange Einarbeitungs- oder Lernphase gilt es auf jeden Fall zu vermeiden.

5.2 Entwicklungsumgebung, Architektur und Umsetzung

Um den zu entwickelnden Prototypen einer möglichst großen Gruppe von Nutzern zugänglich zu machen, wird die Programmiersprache Java für die Implementierung genutzt. Sie ist weit verbreitet und weitgehend plattformunabhängig. Es muss lediglich eine Java Virtual Machine für die jeweilige Plattform zur Verfügung stehen. Als Entwicklungsumgebung kommt Eclipse zum Einsatz. Durch sein Plug-in Konzept ist es sehr flexibel und darüber hinaus frei verfügbar.

Die Architektur des Prototypen spiegelt im Wesentlichen die Komponenten des entwickelten Konzepts zur Mustervisualisierung wider: Mustersuche - Musterdarstellung - Musterinteraktion. Diese drei Komponenten sollen unterschiedliche Funktionalitäten bereitstellen. Abbildung 5-1 zeigt die Komponenten der entwickelten Applikation.

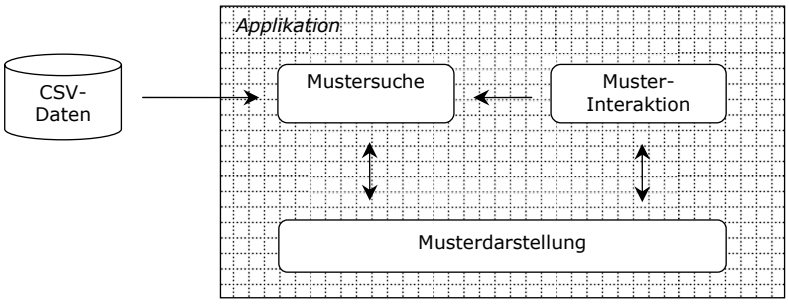

Abbildung 5-1: Komponenten der Applikation

Die Komponente der Mustersuche hat die Aufgabe, die externen Daten in die intern verwendete Datenstruktur der Applikation zu überführen. Die für diese Arbeit verwendete multivariate Zeitreihe liegt im CSV-Format vor. Eine CSV-Datei ist eine tabellenförmig angeordnete Textdatei. Einzelne Werte sind durch spezielle Zeichen getrennt. Für die Speicherung von Multiparameterdaten ist die Nutzung des CSV-Formats weit verbreitet. Die Datenschnittstelle soll deshalb auf dem CSV-Format basieren. Leider ist das CSV-Format nicht immer eindeutig. Als Trennzeichen zwischen den Werten kommen z.B. Kommas oder Semikolons in Frage, häufig sind Zahlenformate sehr unterschiedlich dargestellt. Die Überführung in die interne Datenstruktur wird deshalb mit Hilfe von regulären Ausdrücken durchgeführt. Dadurch ist es möglich, einen hohen Grad an Flexibilität zur erreichen, sodass viele Varianten von CSV-Formaten variabel eingesetzt werden können.

Als interne Datenstruktur für die Implementierung werden verkettete Listen eingesetzt. Verkettete Listen sind für nicht sequentielle Zugriffe langsamer als z.B. Arrays. Listen benötigen aber weniger Speicherplatz. Dies ist insbesondere dann von Vorteil, wenn die Länge der zu speichernden Daten a priori nicht bekannt ist oder sich zur Laufzeit ändert. Die Wahl der Liste als interne Datenstruktur begründet sich durch die zweite Aufgabe der Komponente der Mustersuche: Die Suche nach

Musterpaaren auf den Daten. In 4.2.2 wurde beschrieben, dass die Teilfolge *X* aus der Merkmalreihe selbst extrahiert wird. Dabei werden Teilfolgen *X* stets aus einer einzigen oder aus unmittelbar aufeinander folgenden Ausprägungen gebildet. *X* wird also durch einen sequentiellen Zugriff auf der Merkmalreihe gewonnen. Anschließend wird die Merkmalreihe auf der Suche nach passenden Teilfolgen *Y* sequentiell durchlaufen. Wahlfreie Zugriffe sind nicht erforderlich. Der Geschwindigkeitsnachteil der Listen gegenüber Arrays kommt also nicht zum tragen. Hinzu kommt, dass eine Merkmalreihe nach allen möglichen Musterpaarlängen durchsucht werden soll. Dazu ist die Länge der verwendeten Datenstruktur ständig anzupassen. Listen sind hier gegenüber z.B. Arrays im Vorteil. Die Anpassung der Länge der Listen ist einfacher und schneller realisierbar, als dies in Arrays möglich wäre. Als interne Datenstruktur zur Suche nach Musterpaaren für Arc Diagrams sind Listen also eine geeignete Wahl.

Die Komponente der Musterdarstellung ist verantwortlich für das äußere Erscheinungsbild der Applikation. Sie soll eine einfache Bedienbarkeit und übersichtliche Darstellung gewährleisten. Dafür wird eine graphische Benutzeroberfläche eingesetzt. Die Oberfläche soll möglichst selbsterklärend sein. Auf diese Weise wird eine lange Einarbeitungsphase verhindert. Die Applikation ist in zwei separate Fenster unterteilt – in Bedienungs- und Darstellungsfenster. Im linken Bedienungsfenster nimmt der Nutzer die gewünschten Eingaben vor und stellt Parameter ein. Die Trennung von Mustersuche, Visualisierung und Interaktion ist im Bedienungsfenster beibehalten. Fast alle erforderlichen Eingaben sind mit der Maus über Schaltflächen oder Schieberegler erreichbar. Diese sind zusätzlich durch beschriftete Rahmen gruppiert. Im rechten Darstellungsfenster sollen die ermittelten Ergebnisse graphisch dargestellt werden. Das Darstellungsfenster ist in der Größe variabel einstellbar. Als Voreinstellung wird für jede angegebene Merkmalreihe ein Arc Diagram dargestellt. Die Darstellung wird automatisch skaliert in das Darstellungsfenster eingepasst, um zunächst einen Überblick zu erhalten. Bei Bedarf kann z.B. die Variante der Überlagerungsdarstellung über das Bedienungsfenster ausgewählt werden. Die Umsetzung des Darstellungsfensters richtet sich vor allem nach den Bedürfnissen der Interaktionskomponente. Diese soll nun beschrieben werden.

Das Konzept der Arc Diagrams für multivariate Zeitreihen sieht mehrere Interaktionsmöglichkeiten für den Anwender vor, um ihn bei der Analyse der Zeitreihen zu unterstützen. Diese Interaktionsmöglichkeiten wurden in drei Gruppen zusammengefasst (vgl. 4.4). Um die Realisierbarkeit und den Nutzen dieser Interaktionen aufzeigen zu können, wird mindestens ein Vertreter aus jeder Gruppe implementiert. Insbesondere die geforderte Implementierung eines graphischen und semantischen Zooms und des Scrollen und Panning begründen dabei den Einsatz von Piccolo [Be04]. Piccolo ist ein strukturiertes, graphisches Framework für 2-dimensionale Anwendungen. Es sind vorgefertigte Klassen verfügbar, welche genau die benötigte Interaktion auf der Darstellung (vgl. 4.4) unterstützen. Deshalb wird dieses Framework in der vorliegenden Applikation eingebunden und seine Funktionalität für das

Darstellungsfenster verwendet.

Zum Zweck des graphischen Zoomens bewegt ein Nutzer beispielsweise bei gedrückter rechter Maustaste die Maus nach links für eine Verkleinerung bzw. nach rechts für eine Vergrößerung der Darstellung der Arc Diagrams im Darstellungsfenster. Weitere Interaktionen, wie z.B. das Highlighting, sind im Bedienungsfenster bei Bedarf selektierbar.

5.3 Anwendungsbeschreibung

Dieser Abschnitt soll eine Einführung in die Bedienung des Programms liefern. Da das Programm nach den Anforderungen in 5.1 bewusst einfach aufgebaut ist, kann diese Einführung sehr kurz ausfallen.

Nach dem Start der Applikation sind im ersten Schritt die zu betrachtenden Merkmalreihen anzugeben. Den Maßgaben der Datenkomponente folgend, soll sich an das CSV-Format gehalten werden. Die weiteren Schritte sind in ihrer Reihenfolge unbestimmt. Die Voreinstellung der Applikation verwendet den von [Sh96] vorgeschlagenen Grundsatz: *„Overview first, zoom and filter, then details on demand".* Erfolgt keine weitere nutzergesteuerte Parametrisierung, wird daher automatisch nach allen möglichen Musterlängen der Merkmalreihen gesucht und anschließend alle gefundenen Muster in einer Überblicksdarstellung, skaliert auf die verfügbare Größe des Darstellungsfensters, angezeigt. Je nach Umfang der Daten- und Mustermenge ist die Überblicksdarstellung zunächst gut oder weniger gut interpretierbar. Wurden so viele Muster gefunden, dass eine Interpretation der Darstellung unmöglich ist, kann eine interaktive Selektion der Musterlänge erfolgen, um nur eine Teilmenge aller gefundenen Muster darzustellen. Werden z.B. aufgrund eines sehr großen Wertebereichsumfangs kaum Muster gefunden, kann z.B. ein Nutzer mit Hilfe der Unscharfen Suche zielgerichtet interaktiv Toleranzen für die Suche und Darstellung eines Musters festlegen. Um feine Details der Darstellung anzuzeigen, kommen integrierte Interaktionstechniken zum Einsatz. Das Panning beispielsweise erfolgt durch das Ziehen der Maus bei gedrückter linker Maustaste in der Darstellung. Ein Zoom ist durch analoges Vorgehen mit der rechten Maustaste möglich.

5.4 Ausblick

In der Zukunft kann die Applikation um weitere Funktionalitäten ergänzt werden. Neben der Integration der N-Eck Darstellung wird die Integration einer Exportfunktion als ein wichtiger Ausbauschritt angesehen. Zwei mögliche Exportfunktionen sollen genannt werden. Eine Erste speichert die Arc Diagram Darstellung als Graphik. Dafür soll z.B. das verbreitete PNG-Format zum Einsatz kommen. Diese Möglichkeit ist für Präsentationszwecke gut geeignet.

Eine zweite Möglichkeit ist, die ermittelten Arc Diagrams in das SVG-Format zu exportieren. SVG ist ein Vektorgraphikstandard, welcher vom W3C im Jahr 2001 verabschiedet wurde. [Zo06] betrachtet Möglichkeiten von SVG zur Informationsvisualisierung. Als Vorteile von SVG nennt [Zo06] die Eignung als plattformübergreifendes Austauschformat und die Tatsache, dass viele Viewer in der Lage sind, SVG-Dateien anzuzeigen. Da SVG textbasiert ist, können darüber hinaus bereits erstellte Arc Diagrams problemlos neu eingelesen, aktualisiert und manipuliert werden. Dazu müssen nicht alle Musterpaare neu berechnet werden. Gerade für große Datenmengen, wie die vorliegende multivariate Zeitreihe, stellt dies einen wichtigen Vorteil dar.

Kapitel 6

Schlussbetrachtung

Die Arc Diagram Technik ist zur Visualisierung von Mustern in Zeichenfolgen gut geeignet. Im Verlauf dieser Arbeit wurde die „Arc Diagram" Technik so erweitert, dass auch Muster in multivariaten Zeitreihen visualisiert werden können. Dieses Ziel wurde durch drei Ansätze erreicht.

Die Suche nach Mustern wurde beschleunigt durch den Einsatz einer fallbezogenen Auswahl an Suchalgorithmen. Durch die Unscharfe Suche und die Distanzsuche gelang es, Muster auch in multivariaten Zeitreihen mit großem Wertebereichsumfang zu finden. Darüber hinaus wurde die Integration von mathematisch-statistischen Verfahren zur Suche nach multivariaten Mustern in die Arc Diagrams vorgeschlagen, weil diese Zusammenhänge bislang nicht gefunden werden konnten.

Die Darstellung der „Arc Diagrams" wurde speziell für multivariate Zeitreihen angepasst. Die Überlagerungsdarstellung und die N-Eck Darstellung ermöglichen es, auf Basis der Arc Diagrams neben univariaten auch multivariate Zusammenhänge darzustellen.

Die Integration von Interaktionstechniken in die Arc Diagrams stellt einen wichtigen Bestandteil des vorgestellten Konzepts dar. Diese ermöglichen es dem Nutzer, zielgerichtet in den Such- und Darstellungsprozess einzugreifen.

In dieser Arbeit wurden wichtige Aspekte berücksichtigt, welche es ermöglichen Arc Diagrams auch zur Visualisierung von multivariaten Zeitreihen einzusetzen. Dennoch gibt es weitere Aspekte, welche in dieser Arbeit nicht berücksichtigt wurden. Für weitere Untersuchungen sollte z.B. die 3-dimensionale Anordnung von Arc Diagrams erwogen werden, weil dadurch eine zusätzliche Dimension zur Verschlüsselung der hohen Datenmenge multivariater Zeitreihen zur Verfügung steht. Auch ist es sinnvoll, über weitere Darstellungsvarianten der Arc Diagrams nachzudenken, um beispielsweise den zur Verfügung stehenden Platz noch besser auszunutzen oder Muster noch effektiver hervorzuheben.

Abbildungsverzeichnis

Tabellenverzeichnis

Literaturverzeichnis

[An01] Ankerst, M.: *Visual Data Mining with Pixel-oriented Visualization Techniques.* Workshop on Visual Data Mining. ACM SIGKDD, San Francisco, 2001.

[Ba06] Backhaus, K. et al.: *Multivariate Analysemethoden: Eine anwendungsorientierte Einführung*. Springer Verlag, Berlin u. a., 2005.

[Be04] Bederson, B. et al.: *Toolkit Design for Interactive Structured Graphics.* IEEE Transactions on Software Engineering, 30 (8), pp. 535-546, 2004.

[Be81] Bertin, J.: *Graphics and Graphic Information Processing.* de Gruyter, New York, 1981.

[BF92] Beshers, C.; Feiner, S.: *Automated Design of Virtual Worlds for Visualizing Multivariate Relations.* Proceedings of the 3rd conference on Visualization '92, IEEE Computer Society Press, Los Alamitos, 1992.

[Ch82] Chatfield, C.: *Analyse von Zeitreihen: Eine Einführung.* Teubner Verlag, Leipzig, 1982.

[CH92] Church, K.; and Helfman, J.: *Dotplot: A Program for Exploring Self-Similarity in Millions of Lines of Text and Code.* Proceedings of the 24th Symposium on the Interface, Computing Science and Statistics, 1992.

[ESK97] Encarnaçaõ, J.; Straßer W.; Klein R.: *Graphische Datenverarbeitung 1 und 2.* Oldenbourg Verlag, München, Wien, 1996/97.

[He96] Helfman, J.: *Dotplot Patterns: A Literal Look at Pattern Languages.* Theory and Practice of Object Systems (TAPOS), special issue on Patterns, V2, 1996.

[HR83] Hamori, E.; J. Ruskin: *H-Curves, A Novel Method of Representation of Nucleotide Sequences Especially Suited for Long DNA Sequences*. Journal of Biological Chemistry, 1983.

[KR87] Karp, R.; Rabin, M.: *Efficient Randomized Pattern-Matching Algorithms.* IBM Journal of Research and Development 31, 1987.

[La03] Lang, H.: *Algorithmen in Java*. Oldenbourg Verlag, München, Wien, 2003.

[La07] Lang, H. (2007): *Algorithmus von Karp-Rabin.* http://www.iti.fh-flensburg.de/lang/algorithmen/pattern/karp.htm, 14.01.2007.

[NM99] Noe, S.; Müller, W.: *Visualisierung von molekular-biologischen und genetischen Daten.* Technischer Bericht, TU Darmstadt, 1999.

[OL05] Ohm, J.; Lüke, H.: *Signalübertragung: Grundlagen der digitalen und analogen Nachrichtenübertragungssysteme.* Springer Verlag, Berlin, 2005.

[Sh96] Shneiderman, B.: *The Eyes have it: A task by data type taxonomy of information visualizations.* Proceedings of IEEE Symposium on Visual Languages'96, IEEE Computer Society Press, Los Alamitos, 1996.

[SM00] Schumann, H.; W. Müller: *Visualisierung.* Springer Verlag, Berlin, 2000.

[Sp06] Spence, R.: *Information Visualization*. Addison-Wesley, Harlow u. a., 2006.

[SS99] Schlittgen, R.; Streitberg, B.: *Zeitreihenanalyse*. Oldenbourg Verlag, München, 1999.

[Tu83] Tufte, E.: *The visual display of quantitative information.* Graphics Press, Chesire, 1983.

[Wa02] Wattenberg, M.: *Arc Diagrams: Visualizing Structure in Strings*. Proceedings of IEEE Symposium on Information Visualization, 2002.

[Zo06] Zornow, M.: *Einsatzmöglichkeiten von SVG zur Informationsvisualisierung auf mobilen Endgeräten.* Studienarbeit, Universität Rostock, 2006.

Eidesstattliche Erklärung

Ich erkläre hiermit eidesstattlich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe angefertigt und alle Abschnitte, die wörtlich oder annähernd wörtlich aus einer Veröffentlichung entnommen sind, als solche kenntlich gemacht habe, ferner, dass die Arbeit noch nicht veröffentlicht und auch keiner anderen Prüfungsbehörde vorgelegt worden ist.

Zingst, den 28. Februar 2007

Thesen

1. Es existiert eine Reihe von Verfahren zur Visualisierung von zeitabhängigen Daten. Diese Verfahren beschränken sich zumeist auf die Darstellung der Daten, eine Hervorhebung von Mustern wird bislang nicht in ausreichendem Maße unterstützt.
2. Die Hervorhebung von wiederkehrenden Mustern trägt wesentlich zum Verständnis der in Datenmengen enthaltenen Informationen bei.
3. Die „Arc Diagrams" sind eine gut geeignete Technik zur Visualisierung von Mustern in Zeichenfolgen. Die Technik läuft in zwei Schritten ab. Im ersten Schritt werden Muster durch einen Algorithmus gesucht. Im zweiten Schritt werden gefundene Muster dargestellt.
4. Multivariate Zeitreihen weisen in der Regel einen größeren Daten- und Wertebereichsumfang als Zeichenfolgen auf. Die Mustersuche der „Arc Diagrams" ist für den Einsatz in multivariaten Zeitreihen daher bislang zu langsam und zu unflexibel.
5. In multivariaten Zeitreihen können neben Mustern innerhalb einer Merkmalreihe auch mehrdimensionale Zusammenhänge auftreten. Diese werden von „Arc Diagrams" bislang weder gefunden noch adäquat visualisiert.
6. Ein im Rahmen dieser Arbeit neu entwickeltes Konzept beschleunigt die bisherige Mustersuche und ermöglicht es auf diese Weise, Muster in multivariaten Zeitreihen in akzeptabler Zeit zu finden. Weiterhin kann durch die vorgestellte Unscharfe Suche und die Distanzsuche die Flexibilität der Mustersuche gesteigert werden, welches eine Musterfindung auch in multivariaten Zeitreihen mit hohem Wertebereichsumfang zulässt.
7. Die Musterdarstellung durch „Arc Diagrams" wurde durch eine Reihe vorgestellter, neuer Möglichkeiten zur Verschlüsselung zusätzlicher Informationen und Parameter verbessert. Die entwickelten Varianten der Überlagerungsdarstellung und N-Eck Darstellung erlauben es „Arc Diagrams" darüber hinaus, auch mehrdimensionale Zusammenhänge zu finden und zu visualisieren.
8. Ein neu entwickeltes Konzept zur Integration von Interaktionstechniken gibt dem Nutzer einen größeren Einfluss auf den Such- und Darstellungsprozess der „Arc Diagrams". Dies ist für eine Nutzung der „Arc Diagrams" zur Visualisierung von multivariaten Zeitreihen unter anderem aufgrund des höheren Datenumfangs unerlässlich.
9. Die entstandene Applikation setzt die wesentlich Schwerpunkte des entwickelten Konzepts um. Die Applikation kann daher zur Visualisierung von Mustern in multivariaten Zeitreihen mit Hilfe der „Arc Diagrams" eingesetzt werden.