Aus dem Institut für Signalverarbeitung
der Universität zu Lübeck
Direktor: Prof. Dr.-Ing. Alfred Mertins

Ereignisdetektion mit linearen Modellen und bedingten Zufallsfeldern

Inauguraldissertation
zur
Erlangung der Doktorwürde
der Universität zu Lübeck

Aus der Sektion Informatik / Technik

vorgelegt von
Dierck Emil Otto Matern
aus Lüneburg
Lübeck, 2015

1. Berichterstatter:	Prof. Dr. Alfred Mertins
2. Berichterstatter:	Prof. Dr. Karsten Keller
Vorsitzender der Prüfungskommission:	Prof. Dr. Andreas Schrader
Tag der mündlichen Prüfung:	08. Juni 2016
Zum Druck genehmigt, Lübeck, den	09. Juni 2016

Zusammenfassung

Die Ereignisdetektion ist eine Problemstellung der statistischen Signalverarbeitung. Ein zeitlich abhängiges, statistisches Signal wird dabei auf Unregelmäßigkeiten untersucht. Diese Unregelmäßigkeiten werden *Ereignisse* genannt. Insbesondere ist dabei interessant, dass die Ereignisse vor dem Auftreten vollkommen unbekannt sind; das heißt, es liegen ausschließlich Informationen über den *Normalfall* vor. Dadurch werden die Ereignisse als Abwesenheit des Normalfalls angenommen: sind die Abweichungen zu dem, was als normal angenommen wird, groß, so wird gesagt, dass ein Ereignis beobachtet wurde.

Bedingte Zufallsfelder sind diskrete Markov-Modelle, die auf dem Prinzip der maximalen Entropie beruhen. Dieses Prinzip gibt eine Empfehlung zu Verteilungsfunktionen, die man zur Modellierung verwenden kann: sind alle testbaren Informationen, die einem gegeben sind, in einem Modell integriert, so ist die Verteilung, die am besten die statistischen Daten beschreibt, diejenige mit der maximalen informationstheoretischen Entropie. Wendet man dieses Prinzip auf Markov-Modelle an, so bilden vedingte Zufallsfelder eine optimale Lösung zur Beschreibung von Daten.

Der Kern dieser Arbeit ist es, eine Methodik zu beschreiben, wie bedingte Markov-Zufallsfelder auf das Problem der Ereignisdetektion angewendet werden können. Diese Modelle besitzen Eigenschaften, die in der Ereignisdetektion interessant sind, insbesondere sind sie in der Lage, sehr unterschiedliche Datenquellen in einem geschlossenen Modell zusammenzufassen, sowie mit unvollständigen Daten weiterhin eine Klassifikation ausführen zu können. Bei der Überführung dieser Modelle zur Ereignisdetektion wurde darauf geachtet, dass besondere Eigenschaften des bedingten Markov-Zufallsfeldes erhalten bleiben.

Die Methodik, die im Folgenden beschrieben wird, ist dafür entworfen worden, ein komplettes System zur Ereignisdetektion zur Verfügung zu stellen. Hierunter fällt die Entwicklung eines unüberwachten Trainings von bedingten Zufallsfeldern, die Beschreibung von Markov-Modellen, die die Vorteile von bedingte Zufallsfeldern mit einfacheren Modellen, den Maximum-Entropy-Markov-Modellen, vereint, und die Interpretation von Ergebnissen bei angewendeten Methoden.

Vorwort und Danksagung

Es ist wohl üblich, die Danksagung mit den Worten “hiermit danke ich” anzufangen, darum schreibe ich diesen Satz nur, um nicht in solchen Plattitüden zu verfallen, denn die wären in Summe doch nicht sehr dankbar, nicht persönlich, und würden den wahren Dank nur schmälern. Und dennoch muss, nicht aus Tradition, sondern insbesondere aus seiner Rolle und seiner Person heraus als erstes Prof. Dr. Alfred Mertins einen Dank meinerseits erhalten. Alfred war und ist nicht nur der Leiter des Instituts für Signalverarbeitung an der Universität Lübeck und mein Doktorvater, nicht nur der Drittautor aller wissenschaftlichen Veröffentlichungen, die irgendwie zu dieser Arbeit geführt haben, sondern auch ein stets fähiger Ratgeber, ein freundlicher Chef und eine Persönlichkeit, von denen man gerne mehr im Leben trifft. Darum muss an alle, die nach mir noch bei ihm arbeiten, der Rat erfolgen: Hört zu, wenn Alfred etwas sagt! Er spricht zwar leise und oft mit nicht ganz vollständigen Sätzen, aber mit Inhalt. Und wenn er die Stirn runzelt, dann nicht, weil er etwas nicht versteht, sondern weil er Euch sagen will: Ihr redet vollständigen Blödsinn! Und darum Ohren auf und genau auf Alfred achten.

Alfred ist zwar nicht so direkt in seinen Äußerungen, aber der zweite im Bunde derer, denen ich danken will, ist das doch oft, ohne dass man es ihm verübeln kann: Dr. habil. Alexandru Paul Condurache. Stets der Zweitautor meiner Veröffentlichungen hat er doch noch öfter mir einen Rat gegeben als Alfred, und weil er es immer gut meinte, sein Rat sehr hilfreich war und er auch sonst ein sehr lustiger Zeitgenosse ist, muss ich ihm einfach verzeihen, wie er sich immer an meine Bürotür geschlichen und dann mit Schwung die Türklinke betätigt hat. Wer mit Alex arbeitet, der weiß genau, was ich meine, und warum ich um mehr als die dreieinhalb Jahre gealtert bin, die ich mit ihm zusammen am ISIP war.

Und natürlich muss ich für die tolle Zeit auch meinen anderen leider ehemaligen Kollegen danken. Da wäre Dr. Jan Ole Jungmann, mit dem mich auch eine persönliche Freundschaft verbindet, die über die Zeit des ISIP hinausgereicht hat und noch immer anhält, wie er auch zu allen anderen eine Freundschaft hält - ehrlich, er ist ein sehr freundlicher Zeitgenosse. Leider habe ich noch keinen neuen Kollegen getroffen, mit dem ich das erste Käffchen des Tages zusammen trinken kann, um den Arbeitstag gebührend einzuleiten. Dr. Radoslaw “Radek” Mazur wäre der nächste, mit vielen wenig schmeichelhaften Spitznamen seitens der Studenten versehen, und dennoch ist er ein sehr sympathischer Kerl, wenn man ihn näher kennt. Ich nehme ihm wirklich keine seiner Spitzfindigkeiten übel, und sie waren auch nie boshaft. In der Regel war es sogar

sehr unterhaltsam, wenn er und Alex eine, sagen wir mal, Diskussion über ein beliebiges Thema gestartet haben.

Es gibt noch so viele, die am ISIP in dieser Zeit meine Kollegen waren und die tolle Zeit mit begründet haben und denen ich dankbar dafür bin. Olaf Christ zum Beispiel, der zusammen mit dem anderen Professor des Instituts, Ulrich Hofmann, nach Freiburg gegangen ist, ein Nerd, wie man sie nennt, aber der positiven Sorte, der das Wort bestimmt nicht als Schimpfnamen verstehen wird. Dr. Florian Müller, für den es wohl schwer wird, diesen Dank per Google-Suche zu finden, aber der uns allen als sehr positives Vorbild in Sachen Fleiß und Strebsamkeit hätte dienen können, wenn wir dazu geneigt hätten, uns eines zu suchen. Chen Lei, ein sehr netter Kerl, auch wenn er nur selten mit uns gemeinsam Kaffee getrunken hat, was wohl auf die Sprachbarriere zurückzuführen ist. Und es gab noch so viele mehr, die einfach auch die Stimmung des ISIPs ausgemacht haben, unsere Sekretärin Christiane Ehlers, die gute Seele des Instituts, Matze, Kunal, Yjing, Mehrnaz, und noch so viele andere Doktoranden und Studenten, die man gar nicht aufzählen kann.

Und dennoch muss ich noch zwei weitere hervorheben, auch wenn ich allen dankbar bin: Marco Maaß und Phan Quoc Huy. Die beiden habe ich noch gerade so kennengelernt. Und während bis auf Alfred alle der obigen das Institut verlassen haben oder bald verlassen werden, so sind die beiden zu diesem Zeitpunkt aktive Doktoranden und haben, gewissermaßen, den Staffelstab übernommen und lassen das ISIP weiter leben. Dafür danke ich ihnen und ihren neuen Kollegen, und wünsche ihnen viel Erfolg!

Inhaltsverzeichnis

1 Einleitung

1.1 Problemstellung

Die Ereignisdetektion [4, 44] ist ein praktisch relevantes Problem der Mustererkennung und gehört in den Teilbereich der Zeitreihenanalyse. Unter einer Zeitreihe versteht man hierbei in der Regel ein zeitdiskretes Signal. Dieses umfasst sowohl eindimensionale als auch mehrdimensionale Signale, zum Beispiel Videosequenzen. Dieses Signal soll in zwei Klassen eingeteilt werden: die eine Klasse ist der sogenannte *Normalfall*, die andere das *Ereignis*. Der Normalfall gilt als Regelfall, das heißt, er ist in der Zeitreihe überrepräsentiert und bekannt. Das Ereignis ist stark unterrepräsentiert, das heißt, Beispiele hierfür sind sehr selten innerhalb des Signals; auch in einer großen Menge von Beobachtungen kann es vorkommen, dass das Ereignis selten oder überhaupt nicht beobachtet wird. Die Aufgabe der Ereignisdetektion ist es, dieses Ereignis in dem Signal zu erkennen.

Die Seltenheit des Ereignisses wirft dabei das Problem auf, dass es nur schlecht zu modellieren ist. Durch diese Unterrepräsentation des Ereignisses wird oft angenommen, dass das Ereignis innerhalb des Problems unbekannt ist; das heißt, weder der Zeitpunkt noch die Ausprägung des Ereignisses sind vor dem Eintreten bekannt. Eine längere Beobachtung zum Erlernen dieses Falls ist somit nicht oder nur schwer möglich.

Ferner kann die Variation des Ereignisses in einigen Fällen deutlich größer als die des Normalfalls angenommen werden. Das erschwert es, die Ereignisdetektion durch eine direkte Erkennung durchzuführen, da die Beschreibung des Ereignisses oft nicht umfassend ist. Stattdessen wird eine indirekte Einordnung durchgeführt: wird die Hypothese abgelehnt, dass der Normalfall beobachtet wird, wird angenommen, dass das Ereignis eingetreten ist.

Die Aufgabe der Ereignisdetektion kann demzufolge darauf zurückgeführt werden, Unterschiede vom Normalfall zu erkennen. Ist der Unterschied nach einem gegebenen Maß groß, wird angenommen, dass das Ereignis beobachtet wurde. Im folgenden werden nur Beispiele des Normalfalls verwendet, um dieses Maß zu bestimmen; diese Menge wird *Trainingsmenge* genannt. Dieses ist der Extremfall der Überrepräsentation des

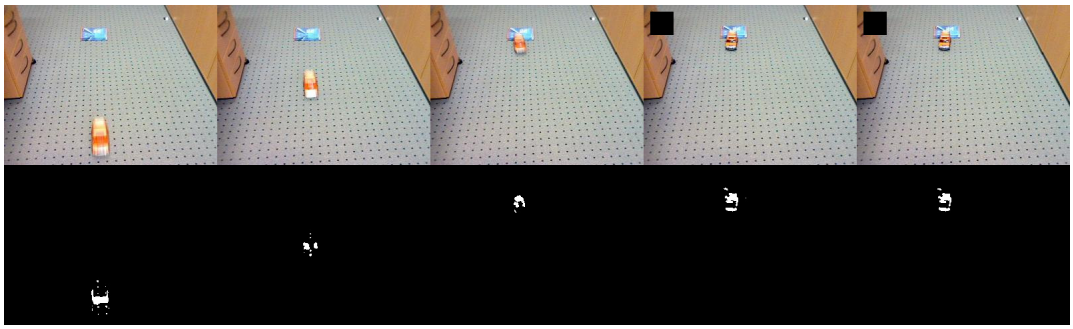


Abbildung 1.1: Beispiel eines Ereignisses: Kollision mit einem Objekt (Einzelbilder mit Markierung). Im unteren Bereich sind die verwendeten Merkmale zu sehen.

Normalfalls, bei dem in der Trainingsmenge das Ereignis nicht beobachtet wird, da es jedoch auch hierfür Beispiele gibt [13], ist dieser Fall ebenfalls relevant.

Zusammengefasst lässt sich Ereignisdetektion somit auf Folgendes zurückführen. Gegeben sei eine Menge von Beispielen, die ausschließlich oder vornehmlich dem Normalfall entsprechen. Die Aufgabe ist es, anhand dieser Beispiele ein Modell des Normalfalls und ein Maß für die Abweichung von diesem zu erstellen. Mit Hilfe dieses Maßes werden neue Messungen dahingegen bewertet, ob sie vom Normalfall abweichen. Sollte die Abweichung groß ausfallen, wurde das Ereignis detektiert.

Mittels der Problemstellung der Ereignisdetektion können viele unterschiedliche Probleme erfasst werden. Überwachungsprobleme bilden hierunter eine Gruppe der prägnantesten Beispiele. Dabei werden Messungen in einer Umgebung getätigt, zum Beispiel werden Areale mittels Kameras, technische Geräte mittels geeigneter Sensoren oder medizinische Faktoren überwacht. Die Messungen werden über lange Zeiträume gesammelt, in denen kein Ereignis auftritt; insbesondere bei der Videoüberwachung können Monate und Jahre vergehen, bevor ein Ereignis stattfindet. Daher kann zwar eine große Datenbank über Messungen bei gegebenem Normalfall angesammelt werden, das Ereignis jedoch ist in dieser Menge nur selten oder gar nicht enthalten.

Die in dieser Arbeit diskutierte Version der Ereignisdetektion ist ein Spezialfall der Ein-Klassen-Klassifikation [44]. Hierbei wird ein Modell für nur eine bekannte Klasse erstellt. Neue Messungen werden dahingehend eingeteilt, ob sie zur bekannten Klasse gehören oder nicht. Bei der Ereignisdetektion wird jedoch immer ein zeitabhängiges Signal angenommen. Dieses bringt vor allem zwei Besonderheiten mit sich. Erstens können Messungen zu einem Zeitpunkt als normal gelten, zu einem anderen hingegen als Ereignis. Zum Beispiel kann das Betreten eines Büroraumes am Tag normal sein,

nachts gilt dieses hingegen als Einbruch. Zweitens können zur Klassifikation statistische Abhängigkeiten zu Messungen genutzt werden, die zu anderen Zeitpunkten aufgenommen wurden. Dadurch werden Informationen über den Kontext für die Klassifikation genutzt.

Klassische Verfahren basieren oft auf künstlichen neuronalen Netzen, *Support Vector Machines*[1] [44] (SVMs), Hidden-Markov-Modelle [65, 88] (HMMs) oder anderen Dichteschätzern [89]. Diese Verfahren sind etabliert, gelten allgemein als effizient und werden oft und für sehr unterschiedliche Verfahren eingesetzt. Um diese Standardverfahren für ein neues Problem zu nutzen, werden die Messwerte für gewöhnlich zunächst prozessiert, das heißt in einen Merkmalsraum transformiert, der topologische Eigenschaften besitzt, auf den das Modell beruht. Insbesondere bei SVMs ist diese Transformation essentieller Bestandteil des Modells [2, 16, 44].

Ein großes Problem bei der Ereignisdetektion wird dabei allerdings in den meisten Fällen ignoriert. Das Ereignis gilt für gewöhnlich als unbekannt. Somit können keine statistischen Eigenschaften des Ereignisses vorausgesetzt werden. Folglich ist nicht bekannt, ob die Form der Transformation für die Ereignisse, die auftreten werden, geeignet ist: Es kann der Fall eintreten, dass die vorher gesetzten Annahmen zu streng für die Detektion eines Ereignisses sind, das heißt, dass durch implizite oder explizite Annahmen die Klassifikationsgüte beeinträchtigt werden kann Insbesondere lassen sich die Annahmen nicht auf Daten des Ereignisses prüfen, sodass hier eine Generalisierung eines Verfahrens beeinträchtigt ist.

Die Alternative ist, die Informationen, die einen in Form von bekannten Messungen gegeben sind, auszunutzen, und anschließend ein Modell zu wählen, dessen Annahmen möglichst gering sind. Eine Lösung dieser Problemstellung ist die Verwendung von Modellen maximaler Entropie. Ein anderer Ansatz ist es, nur datengetriebene Modelle zu nutzen oder möglichst die Annahmen in dem Modell zu begrenzen.

1.2 Eigene Beiträge

Die in dieser Arbeit beschriebenen Verfahren lassen sich in zwei Gruppen einteilen. Die erste Gruppe [52] besteht aus Verfahren, die mittels mehrerer Messungen eine Vorhersage dafür treffen, wie eine folgende Messung erwartet wird. Weicht die Beobachtung von dieser Erwartung ab, so wird angenommen, dass ein Ereignis beobachtet wurde. Dieser Ansatz ist nicht statistisch, es wird demnach keine Verteilung geschätzt; implizite statisti-

[1] Bei einigen Verfahren ist es auch im Deutschen üblich, den englischen Begriff zu verwenden.

sche Annahmen werden dadurch abgeschwächt, dass die Vorhersage nur abschnittsweise getätigt wird und somit keine globale Verteilung notwendig ist. Dieses Vorgehen ist explizit dafür entwickelt worden, die Probleme in den Annahmen zu reduzieren, auch wenn sie nicht vollständig vermieden werden.

Die zweite Gruppe [48–51] beinhaltet Verfahren, die auf den sogenannten *bedingten (Markov-) Zufallsfeldern* (engl. *conditional random fields* [41], CRFs) beziehungsweise *Maximum-Entropy-Markov-Modellen* [54] (MEMMs), siehe Abbildung 1.5, basieren. Es handelt sich bei diesen Modellansätzen um datenabhängige Markov-Zu fallsfelder [22, 28, 31, 41, 65]. Der Unterschied zwischen ihnen liegt insbesondere im Aufbau des Zufallsfeldes.

Um einen allgemeinen Überblick zu geben, werden in Kapitel 2 zunächst übliche Verfahren zur Ereignisdetektion vorgestellt. Ferner werden unterschiedliche Problemstellungen genauer behandelt, die zur Ereignisdetektion gezählt werden können. In Kapitel 3 wird anschließend ein neues Verfahren zur Ereignisdetektion vorgestellt. Dieses Modell basiert auf der Idee der linearen Vorhersage, die Zusammenhänge innerhalb kurzer Abschnitte betrachtet.

In Kapitel 4 werden die Grundlagen für die danach folgenden Verfahren besprochen, das sind CRFs und MEMMs. Diese beiden Konzepte werden sowohl als Markov- als auch logarithmisch-lineare Modelle [69] betrachtet. Eine Vereinheitlichung zu einer echtzeitfähigen Auswertung befindet sich in Abschnitt 4.6.

Bei diesen Verfahren handelt es sich um hybride Markov-Modelle [22]. Dabei werden die Abhängigkeiten der Zufallsvariablen des Modells in einem Graphen dargestellt; eine eingehende Kante zeigt an, von welchen Variablen jede Variable abhängig ist. Bei Hybrid-Graphen erlaubt man gegenseitige Abhängigkeiten, sodass sie nur gemeinsam bestimmt werden können. Ein Vergleich von gerichteten, ungerichteten und hybriden Graphen ist in Abbildung 1.4 zu sehen. Ungerichtete Kanten werden durch Geraden angezeigt, gerichtete durch Pfeile. Hybride Graphen erlauben deutlich komplexere und dadurch fähigere Strukturen, wie in den folgenden Kapiteln dargestellt wird.

Die CRFs werden in Kapitel 5 für die Ereignisdetektion angepasst. Dieses Modell besitzt gegenüber klassischen CRFs einige Neuerungen. Für gewöhnlich werden CRFs mittels einer Merkmalssequenz und einer dazugehörigen sogenannten *Zustandssequenz* trainiert. Das CRF wird derart gestaltet, dass es mit denselben Messungen dieselbe Sequenz als Ausgabe liefert. Dafür müssen für übliche CRFs sämtliche möglichen Elemente der Zustandssequenz in der Trainingsmenge vorhanden sein.

In dieser Arbeit wird das CRF um die Möglichkeit erweitert, auch auf ein unbekanntes

Element, das heißt das Ereignis, zu trainieren. Die übrigen Elemente der Zustandssequenz beschreiben hierbei den Normalfall. Zusätzlich wird gezeigt, wie auf eine bekannte Zustandssequenz verzichtet werden kann. Diese Modifikationen erweitern die Anwendbarkeit von CRFs deutlich, da hier eine weitere Einschränkung des klassischen Ansatzes aufgehoben wird.

In Kapitel 6 werden anschließend Experimente beschrieben. In diesen wurde in unterschiedlichen Versuchen gezeigt, dass die vorgestellten Modelle in vielen möglichen Situationen angewendet werden können. Die große Variation der Anwendungsgebiete ist ein wichtiges Argument für die hier vorgestellten Algorithmen. Die Verfahren und Ergebnisse wurden in [48–52] veröffentlicht.

Die wissenschaftlichen Beiträge dieser Arbeit sind das Vorstellen eines Maßes zur Analyse von Zeitreihen für die Ereignisdetektion, die Überführung des Normalfalls in die Darstellung eines CRFs, die Entwicklung eines Trainingsverfahren für CRFs mit einem Zustand ohne Beispiele, die automatische Einteilung eines Datensatzes in unterschiedliche Segmente mittels eines CRFs, die Verbindung zwischen MEMM und CRF in Form eines Hybrid-Graphen und die Anwendung der entwickelten Algorithmen auf reale Datensätze. Da die verwendeten Methoden auf andere Daten respektive andere Merkmalstransformationen angewendet werden als klassische Verfahren, wurde auf einen Vergleich mit Hinsicht auf Überlegenheit in den meisten Fällen verzichtet.

1.2.1 Ereignisdetektion mittels linearer Vorhersage

Bei den Verfahren, die im Folgenden diskutiert werden, ist es die Voraussetzung, den Normalfall erkennen zu können, um ein Ereignis zu erkennen. Das schließt unterschiedliche Merkmale des Normalfalls mit ein. Eines davon, das in vielen Verfahren ignoriert wird, ist die zeitliche Abfolge der Beobachtungen.

Sind sowohl die möglichen Ausprägungen als auch die zeitliche Abfolge bekannt, kann oft eine neue Messung vorhergesagt werden, das heißt, man stellt eine Schätzung über sie an, bevor die Messung getätigt wurde. Eine große Abweichung von dieser Vorhersage ist somit ein Ereignis.

Eine Möglichkeit, eine solche Vorhersage zu treffen, ist die mittels eines linearen Modells [39, 43, 66]. Hierbei wird eine neue Messung als eine gewichtete Summe von vorherigen Messungen und einem Fehlerterm betrachtet [73, 82]. Als Vorhersage wird der Erwartungswert dieser Methode verwendet. Dieses Vorgehen hat den Vorteil, dass zur Bestimmung des Modells nur die Gewichte bestimmt werden müssen. Dies kann in

dem Sinne geschehen, dass für eine Reihe an Messungen, also die Trainingsdaten, eine Vorhersage möglichst präzise ist.

Ein mögliches Maß ist die Summe der Quadrate der Fehler: hierfür werden die Gewichte derart gewählt, dass für die Trainingsmenge das Quadrat der Abweichung zwischen der Vorhersage und einem tatsächlichen Messwert minimal ist [73, 82]. Das Filter, das für diese Messwerte eine Vorhersage tätigt, wird im Folgenden *Prädiktor* genannt. Mit Hilfe von Prädiktoren werden Vorhersagen über zukünftige Messwerte getätigt; ist diese Vorhersage gut, so wird der Normalfall angenommen, ist die Abweichung groß, das Ereignis.

Um die Einfachheit des Modells der linearen Vorhersage mit komplexeren Szenarien zu kombinieren, in denen die Annahmen des Modells nicht global gelten, wird in dieser Arbeit eine Kombination mehrerer solcher linearen Vorhersagen genutzt. Insbesondere erlaubt diese Methode es, dass sehr unterschiedliche Prädiktoren für unterschiedliche Segmente gute Ergebnisse liefern können.

Es handelt sich bei dem Verfahren somit um einen *Multi-Filter-Ansatz*. Diese Filter werden in ein geschlossenes Modell gesetzt, die Ausgabe erfolgt für alle Filter gemeinsam. Dabei wird sowohl der individuelle Fehler als auch die Häufigkeit, mit der ein Prädiktor gute Vorhersagen tätigt, berücksichtigt. Das Ergebnis ist ein skalarer Wert, der mit einem Schwellwert verglichen werden kann. Für diesen Wert werden sowohl zeitliche Zusammenhänge als auch sinnvolle Bereiche der Messwerte gemeinsam bewertet.

Das Modell wird im Detail in Kapitel 3 beschrieben. Es folgt der klassischen Vorgehensweise der Ereignisdetektion, das heißt, es ist eine Methode, die direkt mit anderen verglichen werden kann, da sie dieselben Ereignisse detektiert, das sind Abweichungen von beobachteten Werten. Daher wird dieses Verfahren auch in den Experimenten in Abschnitt 3.3 mit einem klassischen Modell, einem HMM, verglichen [44]. Das HMM wurde gewählt, da es für das gesetzte Szenario die besten möglichen Ergebnisse verspricht. Um die Aussagen zu verallgemeinern, werden in einer Simulation auch Vergleiche zu einem Gauß'schen Mischmodell (GMM), das heißt ein Mischmodell aus mehreren Normalverteilungen, die sich in Mittelwert und Varianz unterscheiden, gezogen.

1.2.2 Markov-Modelle zur Ereignisdetektion

Markov-Modelle [22, 31, 41, 44, 54, 65] sind effektive Methoden zur Beschreibung von komplexen Daten. Es sind statistische Modelle, das heißt, ein zentraler Bestandteil dieser Verfahren sind Zufallsvariablen. Diese Zufallsvariablen sind abhängig voneinander und

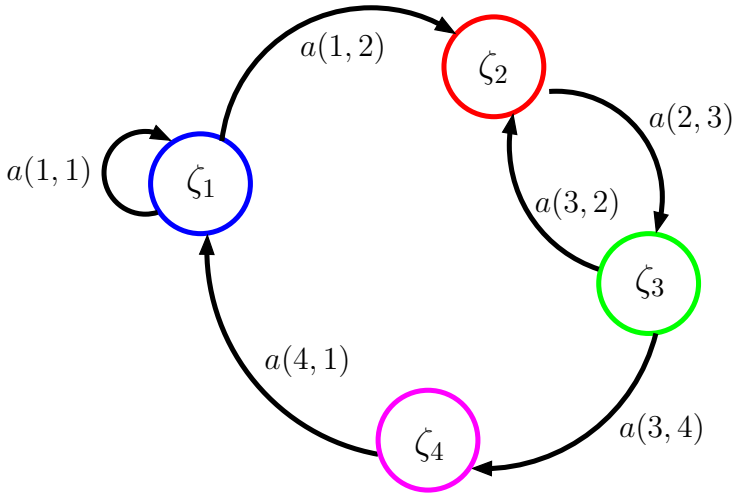


Abbildung 1.2: Beispiel eines Transitionsgraphen eines Markov-Modells [65]. Die Farben (mögliche Zustände des Markov-Modells) werden durch Kreise dargestellt, die möglichen Übergänge durch Pfeile. Das Modell nimmt zu einem festen Zeitpunkt stets eine der Farben an. Im Wechsel zum nächsten Zeitpunkt wechselt das System von einer Farbe in die nächste mit einer gegebenen Wahrscheinlichkeit, hier durch $a(\,\cdot\,,\,\cdot\,)$ notiert.

bilden so ein Netzwerk, das es erlaubt, die Zusammenhänge komplexer physikalischer Systeme besser zu beschreiben, als es ohne solche Verbindungen möglich wäre.

In den Verfahren, die später in dieser Arbeit besprochen werden, werden die Zufallsvariablen in zwei Klassen unterteilt: eine Klasse ist eine statistische Betrachtung der Merkmale, die für die Analyse verwendet werden. Diese werden im Folgenden *Beobachtungen* genannt, da sie durch eine Messung observierbar sind. Ein Messwert ist eine Realisierung der Beobachtung. Durch diese Notation kann von einer Beobachtung zu einem Zeitpunkt gesprochen werden, bevor sie gemessen wurde; dieses Vorgehen ist insbesondere für Vergleiche unterschiedlicher Markov-Modelle hilfreich.

Die zweite Klasse, die *Zustände*, sind dem jeweiligen Modell eigen, das heißt, durch sie werden unterschiedliche Modelle beschrieben. HMMs und CRFs bestehen aus diesen beiden Klassen; ein sehr wichtiger Unterschied ist, dass bei HMMs jedem der Zustände eine Verteilung der Beobachtungen zugeordnet ist, während bei CRFs die Beobachtungen als gegeben angenommen werden und ihre Verteilung nicht modelliert wird. Fehler, die durch ungültige Annahmen der Modellierung entstehen, werden somit vermieden.

Schon an diesem grundliegenden Unterschied zwischen den Modellen zeigt sich auch der Vorteil der Notation: es ist möglich, ein Netzwerk von Beobachtungen und Zuständen im Allgemeinen zu erläutern, wobei deutlich die observierbaren und die bestimmten

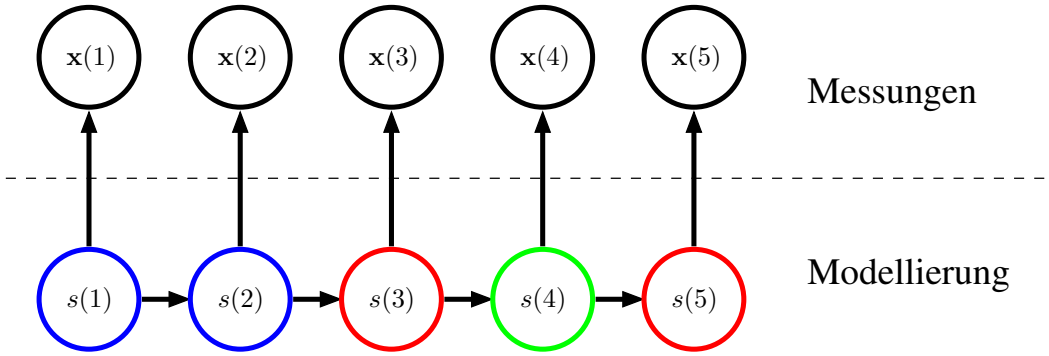
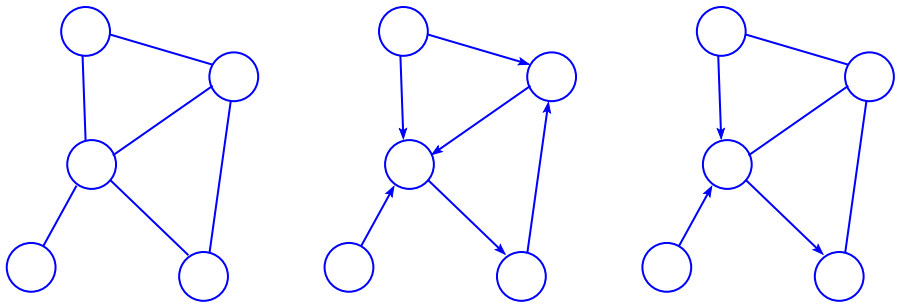


Abbildung 1.3: Abhängigkeitsgraph eines HMMs, das eine Markov-Kette bildet. In dieser Darstellung werden statistische Abhängigkeiten deutlich. Insbesondere ist sichtbar, dass eine Messung in diesem Modell nur von vorherigen Messungen abhängt und diese Abhängigkeiten ausschließlich über das Markov-Modell beschrieben wird.

Zufallsvariablen unterschieden werden können. Eine Beobachtung, die an einem festen Punkt in der zeitlichen Abfolge getätigt wird, hat Einfluss auf die benachbarten Zufallsvariablen. Dieses Netzwerk kann somit im Vorfeld geplant werden, bevor die tatsächliche Messung stattfindet. Nach dieser Messung sind diese Knoten bestimmt, daraus folgt ebenfalls eine Konkretisierung der Zustände. Um diesen Unterschied zwischen der Modellierung und einer Instanz zu verdeutlichen, wird in dieser Arbeit zwischen einer Beobachtung, die an einem Zeitpunkt geplant ist, und ihrer Instanz, dem Messwert, unterschieden, auch wenn dieses ansonsten nicht in jeder Veröffentlichung zu diesen Modellen der Fall ist.

Die Markov-Modelle, die in dieser Arbeit besprochen werden, sind diskret sowohl in der zeitlichen Abfolge als auch in den möglichen Zuständen. Aufgrund letzterer Eigenschaft existiert auch der Begriff *finite state model*, das heißt ein Modell mit endlichen Zuständen [54]. Jeder der Zustände nimmt in einer Realisierung eine der möglichen Ausprägung an. Um die Zustände von den Ausprägungen zu unterscheiden, wird im Folgenden oft von *Färbungen* gesprochen. Eine Instanz eines Zustands ist somit eine Farbe. Ein Zustand ist an einen Zeitpunkt gebunden und statistisch, eine Farbe kann von mehreren Zuständen angenommen werden. Dadurch existiert auch bei den Zuständen eine deutliche Unterscheidung zwischen den statistischen Zufallsvariablen und ihren Instanzen. Diese Notation ist an die Färbung der Graphentheorie angelehnt, allerdings handelt es sich bei den Modellen in dieser Arbeit nicht um ein Färbungsproblem [6].

Eine sehr häufig anzutreffende Darstellung eines HMMs mit einer endlichen Anzahl von möglichen Zuständen ist die mittels eines Transitionsgraphen [65]. Ein Beispiel für



(a) Ungerichteter Graph (b) Gerichteter Graph (c) Hybrider Graph

Abbildung 1.4: Vergleich eines ungerichteten Graphen (a), eines gerichteten Graphen (b) und eines hybriden Graphen (c), der sowohl gerichtete als auch ungerichtete Kanten enthält.

einen solchen befindet sich in Abbildung 1.2. Übergänge mit der Wahrscheinlichkeit 0 wurden hierbei aus Gründen der Übersichtlichkeit nicht mit dargestellt. Dieser Transitionsgraph beschreibt die Übergänge von einer Farbe in die nächste, bietet demnach eine von der Zeit unabhängige Betrachtung des Modells. Ein solcher Graph lässt sich für alle in dieser Arbeit besprochenen Markov-Modelle erzeugen. Anhand dieses Graphen wird die Wahrscheinlichkeit dafür dargestellt, dass eine Farbe einer anderen folgt. Diese Übergangswahrscheinlichkeit charakterisiert das Markov-Modell.

Übergangsgraphen sind üblicherweise gerichtete und gewichtete Graphen. Jedes Gewicht einer Kante entspricht der Wahrscheinlichkeit, dass nach einer Farbe eine andere oder dieselbe Farbe beobachtet wird. Hierbei gilt die Nebenbedingung, dass die Summe der ausgehenden Kanten gleich 1 ist; das kann derartig interpretiert werden, dass fast sicher eine Messung stattfindet. Ist die Wahrscheinlichkeit, dass eine Farbe mehrfach hintereinander beobachtet wird, größer als 0, so ist eine entsprechende Kante im Graphen vorhanden (in Abbildung 1.2: Kante $a(1, 1)$).

Im Fall eines HMMs ist jeder Farbe eine Verteilung zugeordnet, aus denen die Messungen gezogen werden können. Auf diese Art lässt sich ein Ereignis bestimmen: ist die durch die Verteilung gegebene Likelihood eines gemessenen Merkmals gering, so wird angenommen, ein Ereignis beobachtet zu haben.

Gleichzeitig besitzen Markov-Modelle die Möglichkeit, die intuitive Klassifikation von Situationen darzustellen. Jeder Farbe kann eine atomare Aktion zugeordnet werden, zum Beispiel in einem Überwachungsszenario, dass eine Person den Raum betritt oder mit einem bestimmten Gegenstand interagiert oder ihn verlässt. Diese atomaren Aktionen werden sequentiell durchgeführt, das heißt, dass zuerst die Person den Raum betreten muss und erst danach alle Aktivitäten im beobachteten Raum beendet, wenn sie ihn

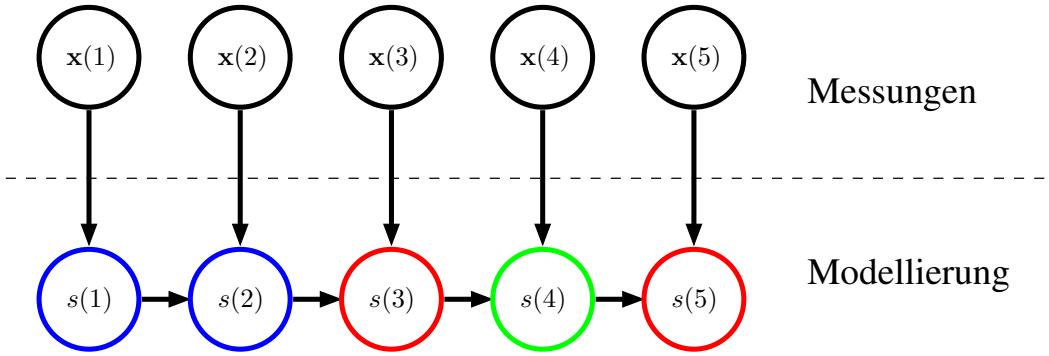


Abbildung 1.5: Abhängigkeitsgraph eines MEMMs. Im Vergleich zu einem HMM fällt auf, dass die Zustände von den Beobachtungen abhängen, nicht umgekehrt. Diese topologische Änderung hat grundlegende Auswirkungen auf die Anwendbarkeit des Modells.

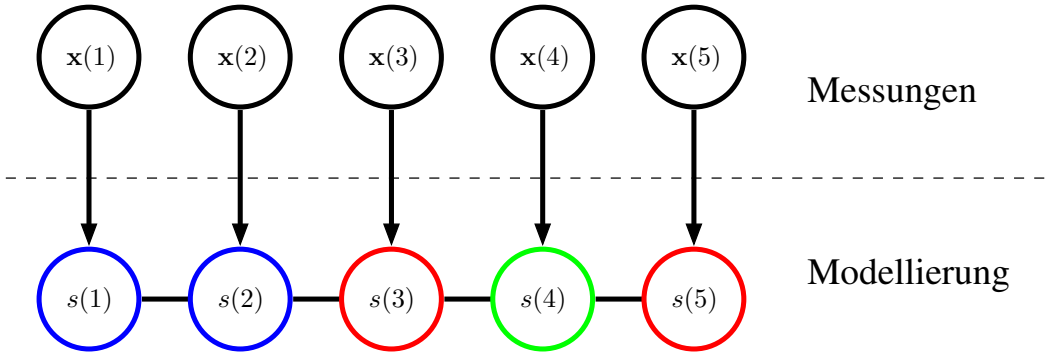


Abbildung 1.6: Abhängigkeitsgraph eines linearen CRFs. Anders als das MEMM bilden die Zustände des CRFs einen ungerichteten Graphen. Dadurch verteilt sich die Information bidirektional im Graphen, die Farbe eines Zustandes hängt sowohl von vorherigen als auch von folgenden Zuständen ab.

anschließend verlässt. Durch die Einteilung in diese atomaren Aktivitäten wird eine Kette erzeugt, die der kompletten Handlung entspricht. Diese Kette lässt sich mittels Markov-Modellen darstellen. Darum werden im Folgenden insbesondere *Markov-Ketten* [65] betrachtet. Ein Beispiel für ein solches Modell ist in Abbildung 1.3 dargestellt. Bei dieser Darstellung wird die Abhängigkeit der zeitdiskreten Zufallsvariablen und der Messungen gezeigt. Dieser Graph ist für die weiteren Verfahren nützlich, weswegen er gegenüber anderen Darstellungen bevorzugt wird.

Als Vereinfachung wird angenommen, dass zu einem Zeitpunkt n der Zustand $s(n)$ vorliegt, und der zeitliche Abstand zu $n-1$ und $n+1$ gleich ist. Dadurch sind die Zeitpunkte, an denen die Zustände definiert sind, immer im gleichen Abstand zueinander. Um zuzulassen, dass eine Aktion eine längere Zeit einnehmen kann, wird erlaubt, dass

mehrere benachbarte Zustände dieselbe Farbe annehmen. Diese Einschränkung ist für die Darstellung der Modelle hilfreich, allerdings nicht notwendig, das heißt, die Modelle sind ebenfalls mit nicht äquidistanten Zeitpunkten anwendbar.

Die Ereignisdetektion mittels eines Markov-Modells kann derart gestaltet werden, dass die Markov-Kette auf Irregularitäten untersucht wird. Werden zum Beispiel Aktionen nicht in bekannter Reihenfolge oder nicht in bekannten Zeiträumen durchgeführt, ist dieses als Ereignis zu bewerten. Markov-Modelle bieten demnach die Möglichkeit, natürliche Interpretationen für die Ereignisdetektion anzuwenden, weswegen sie hierfür sehr interessant sind und den Kern dieser Arbeit bilden.

1.3 Notation

In dieser Arbeit werden übliche Notationen bezüglich Vektoren und Matrizen verwendet. Das bedeutet, diese werden durch gerade, hervorgehobene Buchstaben notiert, wobei ein Großbuchstabe eine Matrix oder eine Sequenz von Vektoren beschreibt, ein kleiner Buchstabe einen Vektor. Eine Ausnahme bilden hierbei griechische Buchstaben, mit denen Parametervektoren notiert werden: diese werden nie hervorgehoben, um der Notation zu folgen, die bei den hiesigen Methoden üblich ist, siehe zum Beispiel [28, 41]. Vektoren sind immer Spaltenvektoren, es sei denn, sie sind explizit transponiert.

Wird eine Matrix oder ein Vektor aus Elementen zusammengesetzt, so wird dieses mit eckigen Klammern angezeigt. Somit ist der Vektor aus 1, 2 und 3

$$\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = [1, 2, 3]^\top.$$

Ein Vektor oder eine Matrix kann durch einen Laufindex angegeben werden. Der erste Laufindex zeigt spaltenweise, der zweite zeilenweise den Index an. Somit ist

$$[a(i)]_{i=1}^{3} = [a(1), a(2), a(3)]^\top$$

ein Vektor, und

$$[a(i,j)]_{i,j=1}^{3} = \begin{bmatrix} a(1,1) & a(1,2) & a(1,3) \\ a(2,1) & a(2,2) & a(2,3) \\ a(3,1) & a(3,2) & a(3,3) \end{bmatrix}$$

eine Matrix. Diese Zusammensetzung kann auch einen Vektor respektive eine Matrix aus anderen Vektoren oder Matrizen erzeugen.

Des Weiteren werden als Übergang zwischen logischen Aussagen und Zahlen die Iverson-Klammern [34] verwendet. Das heißt, dass der Ausdruck

$$[\![P]\!] = \begin{cases} 1 & \text{falls das Prädikat } P \text{ wahr ist,} \\ 0 & \text{sonst} \end{cases} \tag{1.1}$$

verwendet wird. Dieser ist insbesondere für CRFs hilfreich und wird auch in anderen Veröffentlichungen über CRFs verwendet [28, 41].

Eine Besonderheit in der Notation sind Graphen, die ebenfalls durch Großbuchstaben beschrieben werden. Die wichtigsten Graphen sind Markov-Felder. Hierbei ist ein nicht hervorgehobener großer Buchstabe ein Symbol für ein allgemeines Markov-Feld, ein hervorgehobener großer Buchstabe impliziert eine Markov-Kette. Diese Betonung auf Ketten wird an dieser Stelle verwendet, da Markov-Ketten für die vorgestellten Methoden die wichtigste Form von Zufallsfeldern sind. Üblicherweise wird der Buchstabe s für Zustände verwendet; eine Farbe wird durch den Buchstaben ζ notiert. Allenfalls für allgemeinere Betrachtung von Graphen, abseits von Markov-Feldern, wird eine andere Notation verwendet; dieses wird explizit angezeigt. Da hierbei eine doppelte Notation zu Matrizen entsteht, wird stets im Kontext von Ketten respektive Matrizen gespochen, sodass eine Verwechslung vermieden wird.

In dieser Arbeit werden vor allem Zeitreihen analysiert. Das bedingt, dass Auswertungen an Zeitpunkten stattfinden. In dieser Arbeit werden wie üblich diese Zeitpunkte durch Ganzzahlen angezeigt, das heißt, $x(n)$ ist der Messwert mit Zeitindex n. Dieser Zeitindex bezieht sich immer auf die aktuelle Sequenz; in zwei Sequenzen, zum Beispiel zwei aufeinander folgenden Zeitreihen, kann derselbe Index n zwei unterschiedliche Zeitpunkte bedeuten, nämlich der n-te Messwert der jeweiligen Zeitreihe. Der erste Zeitindex ist 1. Der Buchstabe n bezeichnet immer eine Messung zu einem diskreten Zeitpunkt. Wird sich auf eine stetige Zeit bezogen, so wird der Index t verwendet.

2 Methoden zur Ereignisdetektion

Auch wenn viele der Methoden, die in dieser Arbeit diskutiert werden, für eine große Variation von Problemen anwendbar sind, ist der zentrale Anwendungsfall die Ereignisdetektion. Hierbei ist zu beachten, dass dieser Begriff unterschiedlich verwendet wird. Was in dieser Arbeit unter Ereignisdetektion zu verstehen ist, wird im Folgenden anhand einiger Beispiele und bekannter Methoden erläutert. Hierbei wird zunächst ein genereller Überblick geliefert; auch wenn später ausschließlich unbekannte Ereignisse betrachtet werden, wird im folgenden Kapitel diese Einschränkung zunächst nicht gesetzt, um den größeren Zusammenhang darstellen zu können.

Oft findet man Methoden zur Ereignisdetektion im Bereich der Überwachungssysteme, zum Beispiel in [12, 33, 59, 70, 88]. Neben konkreten Anwendungen lässt sich dieses Problem abstrakt erfassen, was zur Erklärung der Ereignisdetektion hilft. Bei dem Problem der Überwachung verfolgt man das Ziel, den Normalfall, also reguläre Messungen, und die Ereignisse, die sich insbesondere durch Unregelmäßigkeiten auszeichnen, zu unterscheiden. Jede Messung wird einem der beiden Fälle zugeordnet. Folglich muss zunächst ein Maß oder eine Beschreibung des Normalfalls definiert werden, gegebenenfalls ebenso vom Ereignis.

Die Messwerte können sowohl durch eine Aufteilung der Sensoren an unterschiedliche Messpunkten räumlich differenziert sowie durch Messungen zu unterschiedlichen Zeitpunkten zeitlich differenziert bestimmt werden. Ferner können sie auch im stochastischen Sinne unterschiedlich verteilt sein, beispielsweise durch eine Kombination unterschiedlicher Sensoren. Diese Messungen kann man durch einen zeitabhängigen Vektor $\mathbf{x}(t)$ repräsentieren, wobei jede Komponente des Vektors $x_i(t)$ einen Messwert des Sensors i zum stetigen Zeitpunkt t repräsentiert. Sensoren, die mehrdimensionale Signale wiedergeben, können der Einfachheit halber als eine Anzahl eindimensionaler Sensoren betrachtet werden. Sofern mehrere zeitlich getrennte, gespeicherte Messwerte zu einem festen Zeitpunkt zur Klassifikation genutzt werden, verwendet man eine entsprechende Verzögerung des Messwerts $x(t)$, also $x(t-\tau)$. Eine Kombination beider Fälle ist ebenfalls möglich, also zum Beispiel der Vektor $\mathbf{x}(t) = [x(t), x(t-\tau)]^\top$. Hierdurch können Messwerte ganzer Zeitintervalle bei einer einzelnen Auswertung genutzt

werden, was für eine Zeitreihenanalyse nützlich sein kann. Die konkrete Auswertung hängt direkt vom Problem selbst ab. Dennoch gibt es einige grundlegende Ansätze, diese Daten zu interpretieren.

Oft wird zur Abstraktion von den direkten Messwerten ein mathematisches Modell zur Ereignisdetektion verwendet, zum Beispiel in [38, 59]. Es gibt unterschiedliche Modellierungen für das Problem, den Normalfall von den Ereignissen zu unterscheiden. Im einfachsten Fall steht eine zuvor festgelegte Beschreibung des Normalfalls bereit, was bedeuten kann, dass ein Messwert einen vorher festgesetzten Schwellwert nicht überschreitet. Ein Beispiel ist die Temperaturüberwachung eines Gerätes mit einer kritischen Temperatur. Erreicht die Arbeitstemperatur diesen Wert, werden Maßnahmen eingeleitet; daher ist dieses Ereignis für die Arbeitsweise ein wichtiger Hinweis. In diesem einfachen Fall ist keine weitere Modellierung notwendig. Es können jedoch Fälle eintreten, bei denen eine derart einfache Unterscheidung zwischen Ereignis und Normalfall nicht möglich ist, oder wo das Problem zuvor auf diesen Fall abgebildet werden muss.

Ist eine einfache vorherige Schwellwertbestimmung nicht möglich, wird oft ein Klassifikator entworfen und auf eine Trainingsmenge trainiert [44]. Dabei existieren mehrere Ansätze, in welcher Form die Klassifikatoren entworfen werden können. Sind genügend Beispiele der Ereignisse vorhanden, so kann das Problem der Ereignisdetektion als Zwei-Klassen-Problem aufgefasst werden [7]. Dabei wird eine neue Messung einem der beiden Fälle zugeordnet. Sind mehrere Ereignisse oder Normalfälle zu unterscheiden, so ist dieses ein Multi-Klassen-Problem [27, 71]. Ersteres ist offensichtlich ein Spezialfall des Multi-Klassen-Problems. Dieses setzt zumindest eine Trainingsmenge oder eine Beschreibung des Ereignisses voraus. Typische Lösungen für diese Probleme sind GMMs [89], HMMs [32] und als stetige Alternative hierzu Kalman-Filter [46, 76, 78], künstliche neuronale Netze [1] sowie SVMs [44].

In anderen Fällen wird das Ereignis in der Trainingsmenge als unterrepräsentiert angenommen und daher das Ereignis nicht direkt trainiert [44]. Die Motivation dahinter ist, dass nicht sämtliche Variationen des Ereignisses in die Trainingsdaten mit aufgenommen werden können. Zudem sind Ereignisse oft sehr selten und weisen in einigen Problemstellungen auch eine größere Variation als der Normalfall auf, sodass eine Modellierung des Ereignisses schwierig ist. Der Normalfall kann hingegen in diesen Fällen deutlich besser beschrieben werden. Ein Klassifikator muss unter diesen Annahmen demnach ausschließlich oder überwiegend anhand von Beispielen des Normalfalls trainiert werden. Da die Voraussetzungen für diese Methoden einfacher zu erfüllen sind als in dem

Fall, dass das Ereignis ausreichend genau bekannt ist, ist dieser Fall für die Praxis von besonderer Bedeutung.

In den nachfolgenden Kapiteln werden insbesondere Klassifikatoren verwendet, die den letzteren Ansatz verfolgen. Als Erweiterung kommt hinzu, dass der Normalfall in weitere Subkategorien aufgeteilt wird. Wird der Normalfall detailliert verstanden, ist eventuell eine genauere Unterscheidung zum Ereignis möglich. Damit können auch sehr komplexe Probleme behandelt werden, die eine hohe Praxisrelevanz haben.

Die Anwendungsgebiete der Ereignisdetektion sind mannigfaltig. Offensichtlich sind hierbei Überwachungssysteme ein sehr wichtiger Bereich. Diese Überwachungssysteme selbst können sehr unterschiedliche Fälle umfassen, von Netzwerkkommunikation [25] über akustische Signale [83, 89] bis hin zu Videoüberwachung [7]. Die hier vorgestellten Beispiele sollen jedoch nicht als eine erschöpfende Menge verstanden werden, sondern aufzeigen, wie groß die Spannweite dieser Systeme ist. In Abschnitt 2.1 wird auf unterschiedliche Überwachungssysteme genauer eingegangen. Neben der Überwachung existieren allerdings auch andere Einsatzmöglichkeiten der Ereignisdetektion. Hiermit befasst sich Abschnitt 2.2. In Abschnitt 2.3 werden bekannte Methoden für dieses Problem erläutert.

2.1 Überwachungssysteme

Eines der wichtigsten Einsatzgebiete der Ereignisdetektion sind Überwachungssysteme. Überwachungssysteme dienen der Beobachtung in sehr unterschiedlichen Szenarien. Bekannt ist insbesondere die automatische Videoüberwachung, die durch die immer größer werdende Verbreitung von Videokameras in jüngerer Vergangenheit sehr an Bedeutung gewonnen hat [7, 12, 30, 33, 38, 40, 70, 79, 87].

Neben der Videoüberwachung ist ferner jede Form von Sensoren für die Ereignisdetektion denkbar, die zeitabhängige Werte liefern. Dabei können von einzelnen Sensoren über homogene Netzwerke bis hin zu Sensornetzwerken, die aus sehr unterschiedlichen Sensoren bestehen, verwendet werden [1, 26, 46].

Ferner existieren auch andere Quellen, aus denen Daten für die Ereignisdetektion generiert werden können. Ein Beispiel hierfür ist die Überwachung von Netzwerkkommunikation [27].

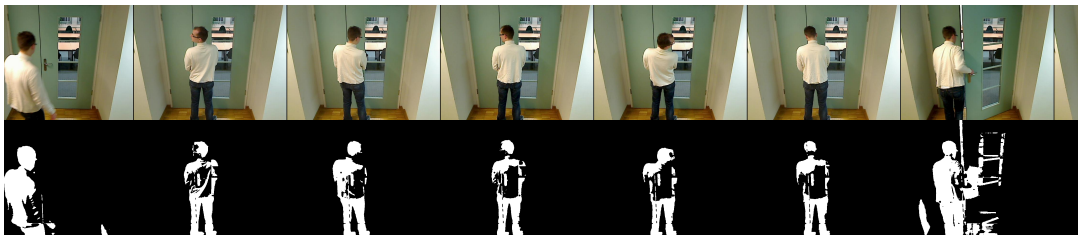


Abbildung 2.1: Beispiel von Überwachungsaufnahmen und mögliche “Merkmale”, die aus den Videosequenzen extrahiert werden können. Die Bilder werden in einer zeitlichen Sequenz aufgenommen. Die für die Entscheidung der Aktion wichtige Information ist in einer einzelnen Aufnahme nicht vorhanden und muss aus dem zeitlichen Zusammenhang genommen werden.

2.1.1 Videoüberwachung

Die Videoüberwachung ist eines der grundlegenden Probleme der Ereignisdetektion und wird zum Beispiel in [12, 33, 42, 49, 70, 79, 90] behandelt. Dieses Problem wird durch die wachsende Anzahl an Überwachungskameras motiviert. Die automatische Auswertung ist notwendig, da die manuelle Interpretation der Videodaten bei sehr seltenen Ereignissen ermüdend ist und zudem die große Anzahl an Kameras den personellen Aufwand erhöht. Eine automatische Auswertung erschwert ferner den Missbrauch der Daten, da Informationen, die zwar zum Normalfall gehören, aber persönlich sind, nicht von Menschen gesichtet werden [85].

Ein System für die Videoüberwachung besteht für gewöhnlich aus einer oder mehreren Kameras [40, 88]. In der Regel sind diese an festen Positionen stationiert, in seltenen Fällen werden sie auch mobil verwendet [12].

Ein typisches Beispiel für die Videoüberwachung, das auch in dieser Arbeit betrachtet wird, ist die Überwachung einer Szene. Das bedeutet, die Kamera nimmt einen festen Raum auf, oder ein Hintergrundmodell wird erstellt und mit den neuen Informationen erweitert; letzteres wird insbesondere bei beweglichen Kameras [12] verwendet.

Desweiteren wird angenommen, dass die Szene nicht leer ist, das heißt, dass Objekte oder Personen im sichtbaren Bereich vorhanden sind. In der beobachteten Szene werden im Normalfall spezielle Aktionen, zum Beispiel übliche Arbeitsschritte oder Bewegungsabläufe, durchgeführt. Diese werden von einer Kamera registriert. Als konkretes Beispiel sei hier ein Wachmann genannt, der in einem per Kamera überwachten Gebiet seiner Tätigkeit nachgeht. Das heißt, er betritt und verlässt die beobachtete Szene, ebenso kontrolliert er eventuell vorhandene Räume oder Gegenstände. Diese Aktionen

sollen von einem Algorithmus als Normalfall klassifiziert werden. Von den Aktionen des Normalfalls sollen insbesondere solche differenziert werden, die ein Dieb oder Einbrecher durchführen würde. Diese sollen als Ereignis klassifiziert werden. Dabei sind die Aktionen des Diebes dem Überwachungssystem nicht vorher bekannt.

Dieses spezielle Beispiel zeigt einige interessante Aspekte der Ereignisdetektion. Im Normalfall werden mehrere unterschiedliche Aktionen durchgeführt. Es kann umständlich sein, diese zuvor separat zu betrachten, also alle Kategorien des Normalfalls im Vorfeld zu erfassen und speziell zu markieren. Daher ist es oft nicht praktikabel, einen überwachten Lernalgorithmus derart anzuwenden, dass er die Unterfälle des Normalfalls berücksichtigt. Ein sogenannter blinder Lernalgorithmus, also ein Algorithmus, der ohne derartig markierte Trainingsdaten auskommt, ist folglich wünschenswert. Allerdings ist es aufwendig, die Aktionen des Wachmanns im Nachhinein zu benennen, sofern so ein blinder Lernalgorithmus verwendet wurde. Da diese Aktionen jedoch gemeinhin dem Normalfall zuzuordnen sind, ist eine genauere Benennung der Aktion auch nicht notwendig für die Überwachung des Bereichs oder im Sinne der Ereignisdetektion.

Ein weiterer interessanter Aspekt ist, dass die Aktionen eines potentiellen Diebes nicht zuvor bekannt sind. Da er jedoch darauf aus ist, seine Aktionen zu verschleiern, kann angenommen werden, dass sie sich möglichst wenig von denen des Wachmanns unterscheiden. Ferner wird er sich nicht durch auffällige Kleidung oder das Zeigen von auffälligen Gegenständen verraten wollen. Dieses wird ihm erleichtert, da er oft nicht direkt seine Aktionen zur Kamera gewandt durchführen muss; das heißt, Kameras können derart installiert sein, dass sie nicht die zu beobachtenden Objekte im sichtbaren Bereich haben, zum Beispiel, um einen größeren Bereich abdecken zu können. Demnach sind Aktionen, das sind Bewegungen und Interaktionen mit Objekten im überwachten Bereich, die einzige Informationsquelle, die zur Detektion von Ereignissen genutzt werden kann. Somit können die Aktionen im Zusammenhang betrachtet werden, um auf diese Art mehr Informationen zu gewinnen; es ist also hilfreich, hier die Ereignisdetektion als Zeitreihenanalyse zu verstehen.

Aus diesen Randbedingungen lassen sich Ansprüche an einen Algorithmus zur Überwachung bestimmen. Insbesondere der Spezialfall, bei dem der Normalfall ebenfalls in Subkategorien aufgeteilt wird, ist sehr nützlich, da elementare Aktionen damit erfasst werden. Eine spezielle Aktion als Ereignis zu bezeichnen, ist nicht sinnvoll, da diese Aktion sehr umfassend definiert werden muss; für multiple Ereignisse erhöht sich das Problem, dass Messwerte unterrepräsentiert sind und somit nur schwer im

Vorfeld trainiert werden können. Des Weiteren ist ein unüberwachter Lernalgorithmus vorzuziehen.

2.1.2 Sensornetzwerke

Ein weiteres Szenario, in dem die Ereignisdetektion wichtig ist, ist die Überwachung von Sensornetzwerken. Dabei kann man zwischen homogenen und heterogenen Netzwerken unterscheiden [58]. In homogenen Netzwerken werden Sensoren derselben Art mit denselben Parametern an unterschiedlichen Positionen des zu überwachenden Bereiches oder Objektes platziert. Auch wenn die Einsatzmöglichkeiten dieser homogenen Netzwerke beschränkt sind, sind sie oft in Verwendung, und die Verbindung der Daten in einem geschlossenen System ist oft einfacher als in inhomogenen Netzwerken.

In einem inhomogenen Netzwerk werden Sensoren unterschiedlicher Typen miteinander verbunden. Hierbei ist zu beachten, dass die Daten, die von den Sensoren aufgenommen werden, sich in ihren statistischen und analytischen Eigenschaften wie Momente und Dimensionen unterscheiden können. Diese Einschränkung kann für viele Methoden zur Ereignisdetektion, bei denen die Verteilung der Daten modelliert wird, problematisch sein. Sind ferner die statistischen Eigenschaften unbekannt, so kann die geschlossene Modellierung für viele Methoden ausgeschlossen sein.

Diese Arbeit befasst sich insbesondere mit Methoden, die für inhomogene Sensornetzwerke geeignet sind. Ein Netzwerk von unterschiedlichen Sensoren, die unter anderem Temperatur, Luftfeuchtigkeit und Bewegungen von Türen messen, ist hierbei ein reell existierendes Beispiel, das aus diesem Grund eine zentrale Bedeutung hat; unter anderem ist in [48] ein solches System verwendet worden.

Temperatur und Luftfeuchtigkeit können für gewöhnlich derartig modelliert werden, dass sie mit Hilfe einer Mischverteilung mehrerer Normalverteilungen ausreichend genau beschrieben werden. Für die Bewegung der Tür wird folgende Modellierung angenommen. Ein Sensor sendet wie oft in einem bestimmten Zeitintervall, zum Beispiel in einer Minute, die Tür geöffnet und geschlossen worden ist. In einem üblichen Haushalt wird eine spezifische Tür selten verwendet, das heißt, die Wahrscheinlichkeit, dass der Sensor keine Betätigung meldet, ist sehr hoch. An mehreren Zeitpunkten am Tag wird die Tür allerdings verwendet. Wird die Tür einmal in einem Zeitintervall betätigt, kann die Wahrscheinlichkeit hoch sein, dass sie innerhalb desselben Zeitintervalls auch mehrmals verwendet wird. Zum Beispiel ist es möglich, dass mehrere Gegenstände innerhalb einer Wohnung von einem Ort zum anderen nacheinander transportiert werden. Eine solche Verteilung ist schwer als Normalverteilung zu modellieren.

Viele der bekannten Methoden, zum Beispiel die am häufigsten verwendete Modellierung eines HMMs [65], nehmen eine Mischverteilung von Normalverteilungen oder direkt eine einzelne Normalverteilung an. Eine Transformation einer Zufallsgröße in eine andere mit einer anderen Verteilung ist unter Umständen möglich.

2.1.3 Andere Datenquellen

Neben Videodaten und Netzwerken von Sensoren sind auch viele andere Datenquellen denkbar. Dieses soll hier nur angeschnitten und nicht erschöpfend aufgezählt werden.

Ein oft betrachtetes Thema ist die Überwachung von Netzwerkkommunikation. Dabei werden die Datenströme an sich analysiert, es werden also keine Netzwerke von Sensoren oder Kameras zur Überwachung eingesetzt [27]. In diesen Datenströmen werden Ereignisse gesucht, zum Beispiel Angriffe von außerhalb. Ein Ereignis ist in der Regel bekannt; das heißt, es wird in dem Datenstrom nach bekannten Mustern gesucht. Da die Angriffe oft komplizierten, aber bekannten Mustern entsprechen und der reguläre Ablauf diffus ist, ist dieses Vorgehen oft der günstigere Fall. Diese Muster können auch in einem überwachten Trainingsalgorithmus gelernt werden.

In medizinischen Zusammenhängen werden Überwachungssysteme oft eingesetzt. Bekannt ist vor allem die Überwachung von biologischen Parametern eines Patienten, zum Beispiel mittels der Elektrokardiographie [9].

Ein Beispiel, das auch näher in den Experimenten in Abschnitt 6.1 betrachtet wird, ist die Detektion eines Kontrastmittels bei Operationen, in etwa bei der Behandlung einer krankhaften Verengung eines Herzkranzgefäßes. Bei einer perkutanen transluminalen koronaren Angioplastie (PTCA) [14] wird diese Verengung mechanisch erweitert. Dazu wird bei der sogenannten Ballondilatation [13–15] ein Ballon zu der Verengung geführt und ausgedehnt. Um diese Position zu finden, sollen die Gefäße mittels Röntgen-Bildern beobachtet werden. Allerdings sind Gefäße auf Röntgenbildern für gewöhnlich nicht sichtbar. Daher wird ein Kontrastmittel verwendet. Das Erreichen dieses Kontrastmittels in dem beobachteten Bereich kann als Problem für die Ereignisdetektion betrachtet werden [51]. Zu Beginn der Prozedur ist kein Kontrastmittel im sichtbaren Bereich, dieses definiert den Normalfall. Unbekannt ist, wie sich das Bild bei Auftritt des Kontrastmittels verändert. Dieses kann als Ereignis aufgefasst werden. In Abschnitt 3.3 wird auf dieses Beispiel genauer eingegangen.

2.2 Alternative Einsatzmöglichkeiten der Ereignisdetektion

Der klassische Einsatzzweck der Ereignisdetektion sind Überwachungssysteme. Aus diesem Grund wurde auf dieses Beispiel ganz besonders eingegangen. Allerdings kann die Ereignisdetektion vielseitig angewendet werden. Insbesondere lassen sich einige klassische Probleme auf die Ereignisdetektion abbilden und damit die Methoden, die für die Ereignisdetektion entwickelt worden, anwenden.

Ein Beispiel wurde bereits in Abschnitt 2.1.3 genannt. Mit Hilfe von Röntgenaufnahmen werden Bereiche observiert, die einer medizinischen Behandlung unterliegen. Neben biologischen Parametern können auch solche beobachtet werden, die durch die Behandlung notwendig sind; in diesem Fall ist es die Anwendung eines Kontrastmittels, um Arterien in den Röntgenaufnahmen sichtbar zu machen.

Prinzipiell lassen sich die entsprechenden Parameter des Bildes messen und bewerten. Der Einfluss der Anwesenheit des Kontrastmittels lässt sich demnach qualitativ messen. Dadurch ist es möglich, zum Beispiel durch eine direkte Schwellwertbildung eine Entscheidung zu treffen, ob das Kontrastmittel im sichtbaren Bereich ist oder nicht.

Ein solches System besteht zum Beispiel darin, Arterien im Bild zu erkennen. Hiermit wurde bereits erfolgreich dieses Problem behandelt [13–15]. Allerdings sind hier mehrere Annahmen notwendig, um die Arterien zu erkennen, so zum Beispiel über die Form dieser Gefäße. Sind diese Annahmen verletzt, zum Beispiel durch die sehr gut sichtbare Anwesenheit einer Kanüle, ist die Qualität dieses Messwertes beeinflusst. Aufwendiger zu messende Eigenschaften müssten also genutzt werden, um dieses Problem mit weniger Annahmen in derselben Form, also durch einfache Schwellwertbildung, zu behandeln.

Eine Alternative hierzu ist, höher entwickelte Klassifikatoren zu nutzen. Die Methoden der Ereignisdetektion gehören hierzu. Dabei wird nur angenommen, dass das Kontrastmittel irgendeinen Einfluss auf gemessene Parameter hat. Dieser Einfluss wird aber als unbekannt angenommen. Dadurch ist das System deutlich fehlertoleranter.

Folglich wird die Aufnahme gestartet, der Messwert wird extrahiert. Das System wird anhand der ersten Aufnahmen trainiert. Nach einigen Sekunden werden dann beliebige Änderungen in der Sequenz detektiert, nicht nur Ausschläge des Wertes in einer bestimmten Richtung. In Abschnitt 6.1 wird dieser Ansatz zur Detektion eines Kontrastmittels an einem Beispiel gezeigt.

Dieses Verfahren gehört zu der Klasse der Detektion von Veränderungen (auf Englisch *Change Detection*) [4, 29]. Diese Detektion kann als ein Spezialfach der Ereignisde-

tektion betrachtet werden. Hierbei werden unbekannte Veränderungen von statistischen Eigenschaften detektiert. Die Ereignisdetektion umfasst allerdings auch den Fall, dass sich der Normalfall in bestimmten Bereichen ändert.

Dieses Beispiel soll dazu dienen, zu zeigen, dass die Ereignisdetektion nicht ausschließlich zur Überwachung eingesetzt werden kann. Andere Probleme lassen sich auf die Prinzipien abbilden. Der Einfachheit halber wird im Folgenden in der Regel von einem Überwachungsproblem ausgegangen.

2.3 Bekannte Methoden der Ereignisdetektion

Zur Ereignisdetektion werden viele unterschiedliche Methoden verwendet. Diese Methoden umfassen probabilistische und nicht-probabilistische Modelle, deskriptive und generative Methoden, ferner auch zeitabhängige und zeitlich invariante Methoden [1, 4, 44, 47]. Die meisten Methoden beinhalten ein Trainingsverfahren. Dieses kann sowohl in einer expliziten Lernphase bestehen, als auch adaptiv in dem Algorithmus selbst verwendet werden.

Ein wichtiges Beispiel einer Methode, die zur Ereignisdetektion verwendet wird, ist die Supportvektormaschine (SVM) [44, 71]. Eine SVM ist ein Klassifikator, der zwei oder mehr Klassen auseinander halten kann. Es handelt sich hierbei um eine erweiterbare, nicht-probabilistische, deskriptive Methode. Prinzipiell mit einer linearen Entscheidungsgrenze verwendet, kann mit Hilfe des sogenannter Kerne (engl. *kernel*) [16, 61] auch eine komplexere Entscheidung getroffen werden.

Bekannte Kerne sind vor allem polynomiale Kerne und solche, die auf Gauß'schen Basisfunktionen basieren [16]. Insbesondere letztere Kerne zeigen gute Ergebnisse bei der Klassifikation von zunächst unbekannten Verteilungen, vor allem bei komplexen Beschreibungen des Normalfalls oder der Ereignisse [2].

Für den Fall der bekannten Ereignisse kann eine Multiklassen-SVM zur Ereignisdetektion verwendet werden. Bei unbekannten Ereignissen ist jedoch eine Einzelklassen-SVM [84] von Vorteil. Hierbei wird die SVM nur anhand des Normalfalls trainiert. Bei dieser Klassifikation wird insbesondere ein Kerne auf Grundlage der Gauß'schen Basisfunktion verwendet.

Künstliche neuronale Netze werden ebenfalls zur Ereignisdetektion verwendet. Insbesondere im Fall der Überwachung haben sie sich als eine praktische Methode erwiesen [44]. Küntliche neuronale Netze sind, wie SVMs, deskriptive, nicht-probabilistische Modelle.

Zeitreihenanalysen werden oft mittels autoregressiver Modelle durchgeführt, ebenfalls für die Ereignisdetektion [20]. Ein solches Verfahren wird in Kapitel 3 verwendet. Hierbei werden lineare Modelle für Zeitreihen auf Ereignisse untersucht. Diese können sowohl als generative als auch deskriptive Modelle interpretiert werden. Die teilweise sehr strengen Annahmen dieser Modelle werden im hiesigen Verfahren derart abgeschwächt, dass sie nur sehr lokal gültig sind. Insbesondere beinhalten autoregressive Modelle eine Stationaritätsannahme, die bei den in Kapitel 3 diskutierten Verfahren nur lokal gültig sein muss.

Einfache generative Modelle basieren oft auf Verteilungsschätzungen, sie sind also in der Regel probabilistische Methoden. Eine oft verwendete Methode ist das Schätzen einer Verteilung mit Hilfe einer Gauß'schen Mischverteilung (engl. *Gaussian mixture model*, GMM) [89]. Hierbei werden die Parameter der Verteilung aus den gegebenen Daten geschätzt.

Eine komplexere Methode der Verteilungsschätzung ist die Schätzung mittels eines generativen Markov-Modells, also insbesondere HMMs oder Kalman-Filter [39]. Ferner gehören ebenfalls die meisten der in dieser Arbeit betrachteten neuen Methoden zu den Markov-Modellen.

2.3.1 Supportvektormaschinen zur Ereignisdetektion

Bei der Ereignisdetektion wird der Normalfall vom Ereignis getrennt. Einer der beiden Fälle liegt stets vor. Daher können auch klassische Verfahren zur Klassifikation verwendet werden. Hierunter fällt insbesondere die SVM.

Der Name der SVM folgt daraus, dass zur Klassifikation die Trainingsmenge auf eine kleinere Menge reduziert wird, die ausreicht, um das Problem zu beschreiben [16]. Mit diesen wird ein Schwellwert bestimmt, der im Fall von zwei Klassen, im aktuellen Problem also dem Normalfall und dem Ereignis, eine lineare Grenze beschreibt, bei der der Abstand zu den dichtesten Vektoren maximal ist [5, 37]. Diese dichtesten Vektoren sind die Supportvektoren. Für nicht linear trennbare Probleme oder Probleme mit überlappenden Klassen sind ebenfalls Lösungen bekannt [5, 16, 44].

Im Falle unbekannter Ereignisse kann der Abstand zu dieser Klasse nicht maximiert werden. Allerdings ist auch eine Ein-Klassen-SVM bekannt [72], also eine SVM, die ausschließlich auf eine Klasse trainiert wird und neue Vektoren dahingehend bewertet, ob diese zu der bekannten Klasse gehören oder nicht. Im einfachsten Fall wird hier ein Kern mit Gauß'schen Basisfunktionen verwendet [72]. Das Verfahren ergibt ein Maß, ähnlich einem Wahrscheinlichkeitsmaß, für den bekannten Fall. Ist das Maß groß, so

ist es wahrscheinlich, dass ein neuer Messwert zum Normalfall gehört. Ist es hingegen klein, wird ein Ereignis angenommen.

Ein Nachteil dieses Verfahren ist, dass es in erster Näherung nicht die Zeitreihe berücksichtigt. Zum Beispiel durch spezielle Transformationen in einen Merkmalsraum, die zeitliche Abhängigkeiten berücksichtigt, kann diese Information integriert werden. In etwa können mehrere Messungen zu einem Merkmalsvektor zusammengefasst werden. Ferner muss dieser Raum eine Metrik beinhalten, sodass der Abstand zum Normalfall, auf dem die Entscheidung beruht, durchgeführt werden kann. Diese Metrik muss zuvor definiert sein. Im Folgenden werden insbesondere Markov-Modelle diskutiert, die eine zeitliche Abfolge in einer natürlicheren Form beinhalten. Ferner ermöglichen CRFs, die später diskutiert werden, die Verwendung nichtmetrischer Merkmalsräume. Eine Kombination aus Markov-Modellen und SVMs wird in [2] diskutiert.

2.3.2 Ereignisdetektion mit Markov-Modellen

Eine der am häufigsten verwendeten statistischen Methoden zur Ereignisdetektion ist die Modellierung mittels eines HMMs [44, 65]. Ein HMM ist ein graphisches, generatives Modell. Mit HMMs können Signale beschrieben werden, auch wenn diese nicht stationär sind. Dennoch sind sie eine Klasse von verhältnismäßig einfachen Modellen.

Ein HMM besteht aus zwei unterschiedlichen Schichten von Zufallsvariablen. Die erste, aus der sich der Name ableitet, ist die “versteckte” Schicht. Sie besteht aus nicht beobachtbaren Zufallsvariablen, den “Zuständen”. Man nimmt an, dass das System sich in einem dieser Zustände befindet, jedoch nicht observierbar ist, in welchem. Das System kann mit bekannten Wahrscheinlichkeiten zwischen den Zuständen wechseln. Bei HMMs nehmen diese Zustände einen Wert aus einer diskreten Menge an; ist die Menge stetig, spricht man vom Kalman-Filter [76].

Die zweite Schicht ist die Ausgabeschicht; das sind die Beobachtungen. Diese Schicht ist observierbar. Man nimmt an, dass die Verteilung dieser Zufallsvariablen statistisch abhängig von der verdeckten Schicht ist.

Ferner sind die Zufallsvariablen dieser Schicht voneinander statistisch unabhängig, wenn die Zustände gegeben sind. Die verdeckte Schicht stellt also die Verbindung zwischen den Beobachtungen dar. Eine graphische Darstellung eines HMMs befindet sich in Abbildung 2.2.

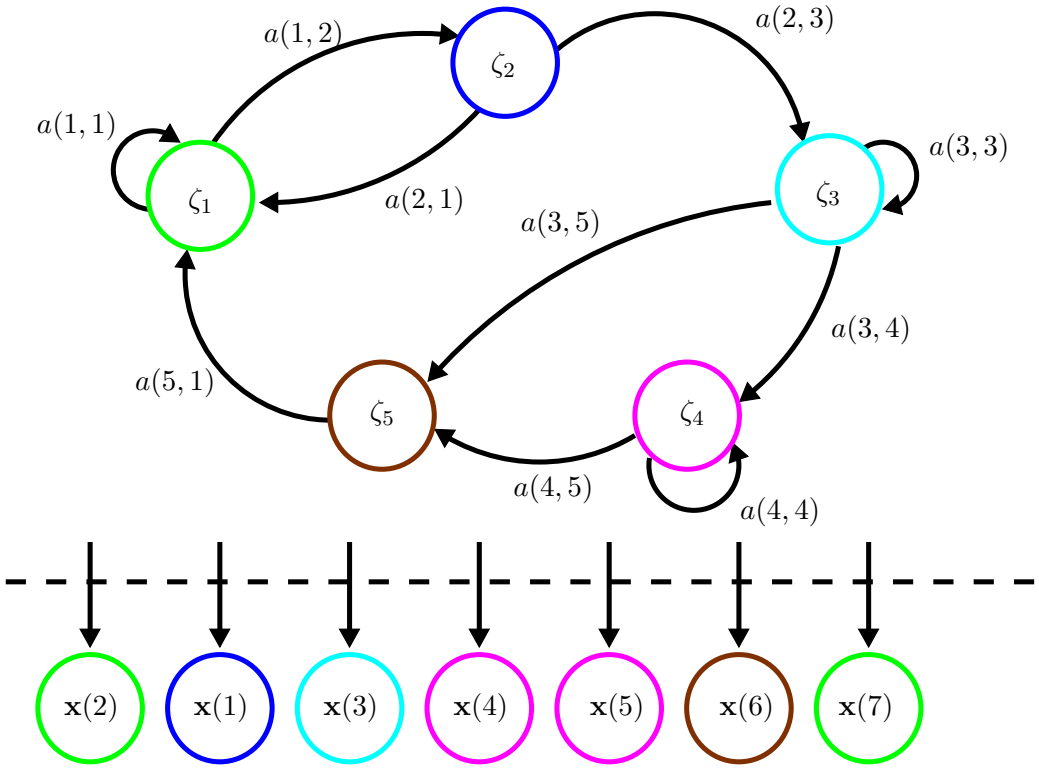


Abbildung 2.2: Darstellung eines Beispiels eines HMMs mit Transitionsgraph und erzeugten Beobachtungen. Diese Darstellung ist üblich für HMMs, zeigt allerdings nicht die Zusammenhänge zwischen den Zuständen und der Kette der Beobachtungen, anders als Abbildung 1.3.

Im Detail modelliert man mit einem HMM eine Verbundwahrscheinlichkeit $p(\mathbf{X}, \mathbf{S})$ von den Beobachtungen $\mathbf{X} = \mathbf{x}(1), \mathbf{x}(2), \ldots$ und den versteckten Zuständen $\mathbf{S} = s(1), s(2), \ldots$ [65]. Für die Verteilung einer Beobachtung $\mathbf{X}$ gilt

$$p(\mathbf{x}(n)|\mathbf{X}\backslash\mathbf{x}(n), \mathbf{S}) = p(\mathbf{x}(n)|s(n)), \tag{2.1}$$

wobei $\mathbf{X}\backslash\mathbf{x}(n)$ sämtliche Beobachtungen ohne $\mathbf{x}(n)$ sind, also

$$\mathbf{X}\backslash\mathbf{x}(n) = \mathbf{x}(1), \mathbf{x}(2), \ldots, \mathbf{x}(n-1), \mathbf{x}(n+1), \mathbf{x}(n+2), \ldots$$

Demnach ist die Verteilung jeder Beobachtung nur von dem Zustand des Systems zum selben Zeitpunkt abhängig. Das bedeutet insbesondere, dass die Information, die eine Beobachtung über die anderen birgt, ausschließlich über das Markov-Modell beschrieben wird.

Das Markov-Modell selbst ergibt sich über die Abhängigkeit der Zustände, das heißt, dass jeder Zustand von anderen statistisch abhängig ist. Üblicherweise wird ein Markov-Modell erster Ordnung angenommen, das heißt, ein Zustand zu einem Zeitpunkt ist nur abhängig von dem Zustand des Zeitpunktes unmittelbar davor,

$$p(s(n)|s(1), s(2), \ldots, s(n-1)) = p(s(n)|s(n-1)), \tag{2.2}$$

wobei $s(1)$ als statistisch unabhängig vom restlichen Graphen angenommen wird.

Häufig wird zur Beschreibung eines HMMs der Transitionsgraph angegeben. Dieser beschreibt die möglichen Übergänge und die damit verbundenen Wahrscheinlichkeiten. Diese Übergangswahrscheinlichkeiten werden im Modell als fest angenommen [65]. Der Übergangsgraph beschreibt aber nur das Markov-Modell an sich; Details zu den Beobachtungen, zum Beispiel deren Verteilungen, sind damit nicht sichtbar.

Ein sehr wichtiger Spezialfall des HMMs ist das HMM, bei dem die Verteilung einer Beobachtung $\mathbf{x}(n)$ bei gegebenen Zustand $s(n)$, von einem GMM beschrieben wird, also

$$p(\mathbf{x}(n)|\mathbf{s}(n) = \zeta_k) \sim \sum_{l=1}^{L} w_k(l) N(\mu_k(l), \Sigma_k(l)), \tag{2.3}$$

wobei ζ_k einer der Farben des HMMs ist. L ist die Anzahl der Normalverteilungen in der Mischverteilung, $w_k(l) \geq 0$, $\sum_{l=1}^{L} w_k(l) = 1$, $\mu_k(l)$ ist der Erwartungswert der l-ten

Normalverteilung für die Farbe ζ_k, $\Sigma_k(l)$ die entsprechende Kovarianzmatrix, und N ist hier das Symbol für die Normalverteilung.

Die Parameter eines HMMs mittels GMM-verteilter Beobachtungen (GHMM) sind demnach die Anzahl der diskreten möglichen Zustände K, also die Menge an Farben $\{\zeta_1, \zeta_2, \ldots, \zeta_K\}$, die Verteilungen der Übergänge $P(s(n) = \zeta_k | s(n-1) = \zeta_j)$ für alle ζ_k und ζ_j, die Initialverteilung $p(s(1))$, die Anzahl der Normalverteilungen in einem GMM L, die Gewichte der Mischverteilungen $w_k(l)$, und die Mittelwerte und Kovarianzmatrizen der einzelnen Normalverteilungen, $\mu_k(l)$ und $\Sigma_k(l)$. K und L sind Designparameter, die für das jeweilige Problem gesetzt werden. Die übrigen Parameter werden in der Regel aus einem Trainingssatz geschätzt. Üblich ist hierzu zum Beispiel der Algorithmus von Baum-Welch [65]. Dieser Trainingssatz beinhaltet nur Beispielsequenzen für die Beobachtungen, nicht für die versteckten Zustände.

Für die Ereignisdetektion bedeutet dies, dass direkt die Messwerte verwendet werden können, um ein GHMM zu trainieren. Im Fall des unbekannten Ereignisses beinhaltet diese Trainingssequenz also nur Messwerte, die den Normalfall beschreiben; weder Werte des Ereignisses noch die Zustandssequenz werden verwendet. Das trainierte GHMM beschreibt demnach auch nur den Normalfall.

2.3.2.1 Datenanalyse mittels Hidden-Markov-Modellen

Die einfachste Methode, mit so einem Modell eine Ereignisdetektion durchzuführen, ist jene über die Likelihood. Das trainierte Modell wird verwendet, um diese Likelihood zu bestimmen. Angenommen, man hat ein auf den Normalfall trainiertes GHMM sowie eine neue Sequenz an Messwerten $\mathbf{X}' = \mathbf{x}'(1), \mathbf{x}'(2), \ldots, \mathbf{x}'(N)$. Ob innerhalb dieser Messwerte ein Ereignis stattgefunden hat, ist unbekannt. Hierfür soll das auf $\mathbf{X}$ trainierte GHMM verwendet werden. Die Formulierungen des GHMMs sind bekannt, zum Beispiel in [65]; im Folgenden wird eine Notation verwendet, die sich insbesondere im Vergleich zu CRFs als nützlich erweist und daher teilweise von den bekannten Notationen unterscheidet.

Es kann mittels der GMMs, die zu den Zuständen gehören, die Likelihood berechnet werden, das ist $p(\mathbf{x}'(n)|s'(n) = \zeta_k)$. Sind die Wahrscheinlichkeiten der Zustände bekannt, also $p(s'(n))$, so gilt für die Wahrscheinlichkeit eines Messwerts

$$p\left(\mathbf{x}'(n)\right) = \sum_k p(\mathbf{x}'(n)|s'(n) = \zeta_k) \cdot P(s'(n) = \zeta_k). \tag{2.4}$$

Die Wahrscheinlichkeit $P(s'(n) = \zeta_k)$ ist abhängig von den Übergangswahrscheinlich-

keiten $P(s'(n) = \zeta_k | s'(n-1) = \zeta_j)$ sowie von den Wahrscheinlichkeiten des Zustandes im vorherigen Schritt $P(s'(n-1) = \zeta_j)$,

$$P(s'(n) = \zeta_k) = \sum_j P(s'(n) = \zeta_k | s'(n-1) = \zeta_j) \cdot P(s'(n-1) = \zeta_j). \tag{2.5}$$

Für $p(s'(n-1))$ gilt derselbe Zusammenhang zum vorherigen Zustand $s'(n-2)$. Ausschließlich für den ersten Zustand $s'(1)$ ist die Verteilung bekannt (das heißt, sie wurde aus den Trainingsdaten $\mathbf{X}$ geschätzt). Ebenso sind die Übergangswahrscheinlichkeiten $P(s'(n) = \zeta_k | s'(n-1) = \zeta_j)$ aus dem Training bekannt.

Folglich lassen sich die Wahrscheinlichkeiten sukzessive bestimmen. Für die Likelihood des ersten Datenwerts $\mathbf{x}'(1)$, also $p(\mathbf{x}'(1))$, gilt

$$p(\mathbf{x}'(1)) = \sum_k p(\mathbf{x}'(1) | s'(1) = \zeta_k) \cdot P(s'(1) = \zeta_k). \tag{2.6}$$

Die Wahrscheinlichkeit des zweiten Zustandes $s'(2)$ ist zunächst abhängig von der Initialverteilung $p(s'(1))$ und der Übergangswahrscheinlichkeit. Für gewöhnlich kann man jedoch auch $\mathbf{x}'(1)$ als Information zur Bestimmung dieser Verteilung verwenden, also man kann die a-posteriori-Verteilung des ersten Zustandes bestimmen mit

$$\tilde{P}(s'(1) = \zeta_k | \mathbf{x}'(1)) = p(\mathbf{x}'(1) | s'(1) = \zeta_k) \cdot P(s'(1) = \zeta_k) \tag{2.7}$$

$$P(s'(1) = \zeta_k | \mathbf{x}'(1)) = \frac{\tilde{P}(s'(1) = \zeta_k | \mathbf{x}'(1))}{\sum_j \tilde{P}(s'(1) = \zeta_j | \mathbf{x}'(1))}, \tag{2.8}$$

wobei $\tilde{p}(s'(1))$ die nichtnormalisierte Verteilung von $s'(1)$ ist.

Die Wahrscheinlichkeit für einen Zustand ζ_k an zweiter Stelle der Zustandssequenz $\mathbf{S}'$ bei gegebener erster Beobachtung ist demnach

$$P(s'(2) = \zeta_k | \mathbf{x}'(1)) = \sum_j P(s'(2) = \zeta_k | s'(1) = \zeta_j) \cdot P(s'(1) = \zeta_j | \mathbf{x}'(1)). \tag{2.9}$$

Man beachte, dass diese Wahrscheinlichkeit berechnet werden kann, bevor die zweite Beobachtung gemessen wurde. Dieses ist für eine Echtzeitanalyse interessant, ferner auch für den Fall, dass die Sequenz $\mathbf{X}'$ nicht begrenzt lang ist.

Dementsprechend wird nicht nur die Information der Übergangswahrscheinlichkeit verwendet, die unabhängig von den Beobachtungen ist, sondern ebenfalls die Information dieser Messungen. Dadurch wird die Information eines Messwertes auch in der

Zustandssequenz des Markov-Modells verwendet. Da die Likelihood späterer Beobachtungen von diesen bedingten Zuständen abhängig sind, wird die Information einer Beobachtung auch für die Likelihood späterer Beobachtungen verwendet.

Die Likelihood für die zweite Beobachtung ist demnach

$$p(\mathbf{x}'(2)|\mathbf{x}'(1)) = \sum_k p(\mathbf{x}'(2)|s'(2) = \zeta_k) \cdot P(s'(2) = \zeta_k|\mathbf{x}'(1)). \tag{2.10}$$

Entsprechend (2.8) kann die (nicht normalisierte) Verteilung für den zweiten Zustand bei gegebenen ersten beiden Beobachtungen bestimmt werden durch

$$\tilde{P}(s'(2) = \zeta_k|\mathbf{x}'(1), \mathbf{x}'(2)) = p(\mathbf{x}'(2)|s'(2) = \zeta_k) \cdot P(s'(2) = \zeta_k|\mathbf{x}(1)). \tag{2.11}$$

Sukzessive kann die Analyse der Datenwerte weitergeführt werden:

$$\begin{aligned} p(\mathbf{x}'(n)|\mathbf{x}'(1), \ldots, \mathbf{x}'(n-1)) = \textstyle\sum_k p(\mathbf{x}'(n)|s'(n) = \zeta_k) \\ \times P(s'(n) = \zeta_k|\mathbf{x}'(1), \ldots, \mathbf{x}'(n-1)), \end{aligned} \tag{2.12}$$

$$\begin{aligned} \tilde{P}(s'(n) = \zeta_k|\mathbf{x}'(1), \ldots, \mathbf{x}'(n)) = p(\mathbf{x}'(n)|s'(n) = \zeta_k) \\ \times P(s'(n) = \zeta_k|\mathbf{x}'(1), \ldots, \mathbf{x}'(n-1)). \end{aligned} \tag{2.13}$$

Da dies zu jedem Zeitpunkt n und für jeden möglichen Zustand ζ_k geschieht, kann die gesamte Verteilung der Zustandssequenz bei gegebenen Messwerten bestimmt werden. Ebenfalls wird die Likelihood der Messwerte bei gegebenen Modell bestimmt.

Zur Ereignisdetektion kann im einfachsten Fall diese Likelihood der Beobachtungen verwendet werden. Das Modell wurde anhand des Normalfalls trainiert, das heißt, die Verteilung der Messwerte und der Übergänge wurden anhand von Trainingsbeispielen geschätzt. Wird in der statistischen Analyse der Normalfall beobachtet, so ist anzunehmen, dass die Likelihood der neuen Beobachtungen, gegeben das Modell, hoch ist, also dass zum Beispiel $p(\mathbf{x}'(n)|\mathbf{x}'(1), \ldots, \mathbf{x}'(n-1))$ größer als ein vorher festgesetzter Schwellwert θ ist. Ist diese Likelihood kleiner als dieser Schwellwert, so wird angenommen, ein Ereignis beobachtet zu haben. Dadurch ist diese Interpretation wieder zurückgeführt auf das Problem der Ereignisdetektion mittels einer Schwellwertbildung.

2.3.2.2 Kalman-Filter

Kalman-Filter werden seltener für die Ereignisdetektion eingesetzt, sollen hier jedoch wegen ihrer Verwandtschaft zu HMMs Erwähnung finden. Das wesentliche Anwendungsgebiet der Kalman-Filter im Bereich der Ereignisdetektion ist die Detektion von

Veränderungen (engl. *change detection*) [4, 29]. Hierbei werden die Änderungen von lokalen statistischen Eigenschaften detektiert. Zum Beispiel können sich innerhalb einer Zeitreihe Mittelwert und Varianz ändern und auf ein anderes lokales Niveau begeben. Die Aufgabe ist nun, diesen Zeitpunkt zu detektieren. Das unterschiedliche Niveau, auch wenn es sich von der Anfangszeit stark unterscheidet, ist nicht zu markieren. In dieser Fragestellung unterscheidet sich die Detektion von Veränderungen zu der hier ansonsten üblichen Ereignisdetektion, die eventuell den Zeitpunkt nicht genau zu bestimmen weiß, aber die unterschiedlichen Niveaus in "Normalfall" und "Ereignis" aufteilt.

Der wesentliche Unterschied des Kalman-Filters zum HMM ist, dass der Zustand stetig und im Allgemeinen mehrdimensional angenommen wird, und nicht diskret eindimensional wie in HMMs [65]. Das heißt, die Zustände sind $\mathbf{s}(n) \in \mathbb{R}^{d_1}$, während die Observablen $\mathbf{x}(n) \in \mathbb{R}^{d_2}$ sind. Der Zusammenhang zwischen diesen beiden Zufallsvariablen ist im einfachsten Falle des Standard-Kalman-Filters

$$\mathbf{s}(n) = \mathbf{A} \cdot \mathbf{s}(n-1) + \nu(n) \tag{2.14}$$

$$\mathbf{x}(n) = \mathbf{B} \cdot \mathbf{s}(n) + \eta(n), \tag{2.15}$$

wobei $\nu(n) \in \mathbb{R}^{d_1}$ und $\eta(n) \in \mathbb{R}^{d_2}$ Rauschterme sind mit

$$\nu(n) \sim N(\mu_1, \Sigma_1),$$
$$\eta(n) \sim N(\mu_2, \Sigma_2).$$

Es wird oft angenommen wird, dass $\mu_1 = \mathbf{0}$ ist, wodurch das Modell bei der Simulation nicht divergiert. Die beiden Matrizen $\mathbf{A} \in \mathbb{R}^{d_1 \times d_1}$ und $\mathbf{B} \in \mathbb{R}^{d_1 \times d_2}$ bestimmen die Übergänge zwischen den Zufallsvariablen.

Die Idee bei der Anwendung des Kalman-Filters ist Folgende: man nimmt an, das beobachtete Signal $\mathbf{X}$ entwickelt sich entsprechend dem Modell und lässt sich demnach auch mit ihm verfolgen. Ähnlich wie beim HMM lässt sich ein Zustand sowohl vor als auch nach einer Beobachtung schätzen. Ändern sich die statistischen Eigenschaften des Signals nicht, so sind die Schätzungen vor und nach der Beobachtung $\mathbf{x}(n)$ ähnlich; das heißt, $\mathbf{x}(n)$ enthält wenig Information. Ist der Unterschied groß, so enthält die Beobachtung viel Information. Eine spontane Veränderung des Signals lässt sich also anhand dieser Information messen. Im englischen Sprachraum wird der Name *Kalman gain* für diese Information der Änderung verwendet [76]. Auf diese Art werden die Veränderungen gemessen, was der Ereignisdetektion entspricht.

3 Ereignisdetektion mittels linearer Vorhersage

Zur Ereignisdetektion kann es sinnvoll sein, die statistischen Eigenschaften der zeitabhängigen Signale zu ermitteln und für die Entscheidung heranzuziehen. Dabei werden Abfolgen von Merkmalen gemeinsam interpretiert, um so eine Aussage über das Ereignis zu liefern. Eine solche Zeitreihenanalyse wird auch im Folgenden diskutiert. Das Verfahren ist ein verhältnismäßig einfaches Modell zur Ereignisdetektion und kann für viele Probleme bereits ohne Anpassungen verwendet werden. Da die Daten ähnlich behandelt werden wie in anderen Methoden, wird dieses Verfahren auch mit Standardmethoden verglichen. Es ist das erste im Rahmen dieser Arbeit neu entwickelte Verfahren.

Lineare Prediktion oder lineare Vorhersage bedeutet, dass über einen Zeitraum mittels einer linearen Kombination von Beobachtungen auf einen oder mehrere zukünftige Messwerte geschlossen wird. Über einen Zeitraum hinweg beschreibt die lineare Vorhersage die Evolution der Daten unter Annahme der Stationarität [66].

Konkret bedeutet dieses Modell, dass die Messwerte gewichtet aufsummiert werden, um einen neuen Wert zu schätzen. Für die Prädiktion um einen Schritt gilt

$$\hat{\mathbf{x}}(n) = \sum_{i=1}^{p} w(i) \cdot \mathbf{x}(n-i), \tag{3.1}$$

wobei p die Länge des Prädiktors ist. Der Prädiktor $\mathbf{w} = [w(i)]_{i=1}^{p}$ bestimmt die Vorhersage des Messwerts $\hat{\mathbf{x}}(n)$. Diese Schätzung erfolgt also ohne Wissen um den tatsächlichen Messwert.

Eine Erweiterung dieses Verfahrens ist die lineare Vorhersage mit Absolutwert:

$$\hat{\mathbf{x}}(n) = \sum_{i=1}^{p} w(i) \cdot \mathbf{x}(n-i) + \mathbf{x}_0. \tag{3.2}$$

Die Gewichte und der Absolutwert werden in den meisten Fällen mit Hilfe eines Trainingssatzes erlernt. Das Trainingsverfahren der Gewichte und des Absolutwertes

kann anhand unterschiedlicher Paradigmen durchgeführt werden. Üblich ist ein Verfahren im Sinne des kleinsten quadratischen Fehler: Hierfür wird der quadratische Prädiktionsfehler

$$\begin{aligned}\epsilon(n)^2 &= \left(\hat{\mathbf{x}}(n) - \mathbf{x}(n)\right)^\top \left(\hat{\mathbf{x}}(n) - \mathbf{x}(n)\right) \\ &= \sum_j \left(\sum_{i=1}^{p} w(i) \cdot x_j(n-i) + x_{0,j} - x_j(n) \right)^2\end{aligned}$$

minimiert, wobei $x_j(n)$ die j-te Komponente des Vektors $\mathbf{x}(n)$ ist. Als Lösung ergibt sich ein geschätzter Gewichtsvektor $\hat{\mathbf{w}}$ und gegebenenfalls ein geschätzter Absolutwert $\hat{\mathbf{x}}_0$. Eine lineare Vorhersage mit diesen Parametern ist eine Annäherung an den Schätzwert in (3.1) respektive (3.2), das ist $\hat{\mathbf{x}}(n)$. Desweiteren ergibt sich ein geschätzter Prädiktionsfehler $\hat{\epsilon}(n)$.

Der Prädiktionsfehler kann ebenfalls zur Ereignisdetektion genutzt werden. Hierfür wird der Prädiktor und gegebenenfalls der Absolutwert auf Daten aus dem Normalfall trainiert. Ist das Signal stationär, sind also die statistischen Eigenschaften wie Erwartungswert und Varianz unabhängig vom Zeitpunkt [5], und lässt sich ferner das Signal mittels (3.1) beziehungsweise (3.2) beschreiben, so ist der Prädiktionsfehler klein [5, 66]. Ist der Prädiktionsfehler hingegen groß, so wird ein Ereignis angenommen.

In praktischen Problemen ist die Annahme der globalen Stationarität zu streng für die Ereignisdetektion. Dieses liegt zum Beispiel daran, dass längere Aktivitäten aus kurzen Aktionen zusammengesetzt sein können: Zum Beispiel kann eine beobachtete Person einen Raum betreten und sich auf einen Stuhl setzen. Dabei seien die Bewegungen ab dem Zeitpunkt des Betretens der Einfachheit halber stetig, das heißt ohne spontane, starke Änderungen. Nimmt man als Merkmal die Position einer Markierung an der Person, so ist diese relativ zum Raum ebenfalls stetig, solange die Person sich zu dem Stuhl hin bewegt, und statisch anschließend. Anstatt einer restriktiven globalen Stationarität, die diese zwei Situationen nicht erfasst, kann man annehmen, dass die Messungen nur über kleinere, lokale Bereiche stationär sind. Das gilt sowohl für die Bewegung hin zum Stuhl als auch für die anschließende Bewegungslosigkeit und den langsamen Übergang zwischen der Bewegung und dem Sitzen.

Diese Annahme der lokalen Stationarität ist schwächer als der globalen; ist letztere erfüllt, gilt ebenfalls diese lokal, anders herum ist dieses nicht zwingend der Fall. Somit kann dieses Modell ebenfalls in dem Fall, dass globale Stationarität vorliegt, angewendet werden.

3.1 Yule-Walker-Gleichungen zur Lösung des linearen Prädiktors

Zur Ereignisdetektion sind in der Regel nur Beispiele des Normalfalls bekannt. Nicht bekannt sind zunächst die Gewichte, mit denen eine lineare Vorhersage getroffen werden kann. Diese werden aus den Trainingsbeispielen geschätzt.

Die hier vorgestellten Lösungen sind nur Spezialfälle der Yule-Walker-Gleichungen [66, 73, 82]. Die Verfahren sind bekannt und werden oft verwendet. Die hier vorgestellte Form kann von üblichen Erläuterungen zu diesem Problem abweichen, ist hingegen für die folgenden Methoden zweckdienlich.

Betrachtet wird zunächst der Fall ohne Absolutwert. Bekannt ist aus (3.1), dass sich ein neuer Wert als Linearkombination der vorherigen Werte schätzen lässt. Der tatsächliche Wert kann hiervon abweichen. Das heißt, der neue Wert $\mathbf{x}(n)$ ist

$$\mathbf{x}(n) = \sum_{i=1}^{p} w(i) \cdot \mathbf{x}(n-i) + \epsilon(n), \tag{3.3}$$

wodurch sich im vorherigen Abschnitt der quadratische Prädiktionsfehler ergibt. Allgemein wird angenommen, dass der Prädiktionsfehler multivariat normalverteilt mit Erwartungswert $\mathbf{0}$ ist, also

$$\epsilon(n) \sim N(\mathbf{0}, \Sigma).$$

Ist das Signal ferner stationär, so lässt sich diese lineare Vorhersage für jeden Messwert ab $\mathbf{x}(p+1)$ bestimmen, also

$$\begin{aligned}
\hat{\mathbf{x}}(p+1) &= \sum_{i=1}^{p} w(i) \cdot \mathbf{x}(p+1-i) \\
\hat{\mathbf{x}}(p+2) &= \sum_{i=1}^{p} w(i) \cdot \mathbf{x}(p+2-i) \\
&\dots \\
\hat{\mathbf{x}}(n) &= \sum_{i=1}^{p} w(i) \cdot \mathbf{x}(n-i).
\end{aligned}$$

Das bedeutet, mit demselben Prädiktor kann jeder Messwert des Signals geschätzt

werden [66]. Sei $\mathbf{X}^p(n) = [\mathbf{x}(n-p), \mathbf{x}(n-p+1), \ldots, \mathbf{x}(n-1)]$ die Matrix des p-elementigen Segments der Sequenz $\mathbf{X}$ von $n-p$ bis $n-1$, dann ist

$$\hat{\mathbf{x}}(n) = \mathbf{X}^p(n) \cdot \mathbf{w}, \tag{3.4}$$

wobei $\mathbf{w} = [w(1), w(2), \ldots, w(p)]^\top$ der Vektor der Gewichte ist. Geschlossen kann die Vorhersage für mehrere Zeitpunkte ebenfalls in Matrix-Notation geschrieben werden. Sei $\mathbf{X}^p_{n,m} = [\mathbf{X}^p(n)^\top, \mathbf{X}^p(n+1)^\top, \ldots, \mathbf{X}^p(m)^\top]^\top$ ein Zusammenschluss aus mehreren Segmenten, und $\mathbf{x}_{n,m} = [\mathbf{x}(n)^\top, \mathbf{x}(n+1)^\top, \ldots, \mathbf{x}(m)^\top]^\top$. Dann gilt

$$\hat{\mathbf{x}}_{n,m} = \mathbf{X}^p_{n,m} \cdot \mathbf{w}, \tag{3.5}$$

wobei $\hat{\mathbf{x}}_{n,m}$ die Schätzung von $\mathbf{x}_{n,m}$ mittels des linearen Prädiktors ist.

Ein guter Prädiktor ist dann gegeben, wenn der Prädiktionsfehler minimal ist. Sind die Beispiele in der Trainingsmenge deutlich mehr als p, ist also die Sequenz der Trainingsmenge länger als der Prädiktor, so existieren auch mehrere Prädiktionsfehler. Für gewöhnlich sollen alle diese Prädiktionsfehler minimiert werden. Minimiert wird dabei oft der Prädiktionsfehler im Sinne des kleinsten quadratischen Fehlers.

Gesucht wird also ein Gewichtsvektor $\hat{\mathbf{w}}$, der eine Vorhersage der Trainingsdaten im Sinne des kleinsten quadratischen Fehlers trifft,

$$\hat{\epsilon}^2_{n,m}(\hat{\mathbf{w}}) = (\mathbf{X}^p_{n,m} \cdot \hat{\mathbf{w}} - \mathbf{x}_{n,m})^\top (\mathbf{X}^p_{n,m} \cdot \hat{\mathbf{w}} - \mathbf{x}_{n,m}) \to \min, \tag{3.6}$$

wobei sich hier der Fehler $\epsilon_{n,m}$ auf das entsprechende Intervall bezieht. Ferner wird hier der geschätzte Fehler $\hat{\epsilon}_{n,m}$ betrachtet, da ein geschätzter Prädiktor verwendet wird; der tatsächlich optimale Prädiktor ist nicht bekannt. Für den quadratischen Fehler gilt

$$\begin{aligned}\hat{\epsilon}^2_{n,m}(\hat{\mathbf{w}})2 &= (\mathbf{X}^p_{n,m} \cdot \hat{\mathbf{w}} - \mathbf{x}_{n,m})^\top (\mathbf{X}^p_{n,m} \cdot \hat{\mathbf{w}} - \mathbf{x}_{n,m}) \\ &= \hat{\mathbf{w}}^\top {\mathbf{X}^p}^\top_{n,m} \mathbf{X}^p_{n,m} \hat{\mathbf{w}} - \hat{\mathbf{w}}^\top \mathbf{X}^{p\top}_{n,m} \mathbf{x}_{n,m} - \mathbf{x}^\top_{n,m} \mathbf{X}^p_{n,m} \hat{\mathbf{w}} + \mathbf{x}^\top_{n,m} \mathbf{x}_{n,m}.\end{aligned}$$

Damit ist der Gradient dieses Fehlers

$$\nabla \hat{\epsilon}^2_{n,m}(\hat{\mathbf{w}}) = 2\mathbf{X}^{p\top}_{n,m} \mathbf{X}^p_{n,m} \hat{\mathbf{w}} - 2\mathbf{X}^{p\top}_{n,m} \mathbf{x}_{n,m}. \tag{3.7}$$

Diesen Gradienten zu $\mathbf{0}$ setzen ergibt

$$\begin{aligned} \mathbf{0} &= 2\mathbf{X}_{n,m}^{p\top}\mathbf{X}_{n,m}^{p}\hat{\mathbf{w}} - 2\mathbf{X}_{n,m}^{p\top}\mathbf{x}_{n,m} \\ \Leftrightarrow \mathbf{X}_{n,m}^{p\top}\mathbf{x}_{n,m} &= \mathbf{X}_{n,m}^{p\top}\mathbf{X}_{n,m}^{p}\hat{\mathbf{w}} \\ \Leftrightarrow \left(\mathbf{X}_{n,m}^{p\top}\mathbf{X}_{n,m}^{p}\right)^{-1}\mathbf{X}_{n,m}^{p\top}\mathbf{x}_{n,m} &= \hat{\mathbf{w}}. \end{aligned} \tag{3.8}$$

Da der quadratische Fehler nicht nach oben begrenzt ist, wird so das eindeutige Minimum bestimmt. Für die Invertierung von ${\mathbf{X}_{n,m}^{p}}^{\top}\mathbf{X}_{n,m}^{p}$ ist es wichtig, dass mindestens p linear unabhängige Vektoren in $\mathbf{X}_{n,m}$ enthalten sind [52, 66]. Das beschränkt m auf $m \geq n + p - 1$, da $\mathbf{X}_{n,m}^{p}$ entsprechend $m - n + 1$ Zeilen besitzt, wobei jede Zeile einem Messwert des mehrdimensionalen Signals entspricht.

Der Fall einer linearen Vorhersage mit Absolutwert lässt sich durch eine Erweiterung der Matrix $\mathbf{X}_{n,m}^{p}$ ableiten. Hierfür erweitert man diese Matrix um den Vektor $\mathbf{1}$, also den Vektor, bei dem jede Komponente den Wert 1 hat, das heißt

$$\mathbf{X'}_{n,m}^{p} = \left[\mathbf{X}_{n,m}^{p}, \mathbf{1}\right].$$

Das letzte Element des geschätzten Vektors

$$\hat{\mathbf{w}}' = \left(\mathbf{X'}_{n,m}^{p\top}\mathbf{X'}_{n,m}^{p}\right)^{-1}\mathbf{X'}_{n,m}^{p\top}\mathbf{x}_{n,m} \tag{3.9}$$

entspricht dem Absolutwert, der für alle Elemente identisch ist. Das ist folglich die Lösung für (3.2), wobei $\mathbf{x}_0 = const \cdot \mathbf{1}$ gilt, also einem Vektor entspricht, bei dem jedes Element gleich ist.

Wird stattdessen für jedes Element ein eigener Wert angenommen, so kann man mit einer anderen Erweiterung den Prädiktor ebenfalls mit demselben Verfahren bestimmen. Hierfür kann man die Matrix

$$\mathbf{Y}^{p}(n) = [\mathbf{X}^{P}(n), \mathbf{I}]$$

verwenden, wobei $\mathbf{I}$ die Einheitsmatrix ist. Äquivalent zu $\mathbf{X}_{n,m}^{p}$ kann man mittels dieser Notation die Matrix $\mathbf{Y}_{n,m}^{p} = [\mathbf{Y}^{p\top}(n), \mathbf{Y}^{p\top}(n+1), \ldots, \mathbf{Y}^{p\top}(m)]^{\top}$ definieren. Mit dieser Matrix ergibt sich ein entsprechender geschätzter Prädiktor $\hat{\mathbf{w}}''$ durch

$$\hat{\mathbf{w}}'' = \left(\mathbf{Y}_{n,m}^{p\top}\mathbf{Y}_{n,m}^{p}\right)^{-1}\mathbf{Y}_{n,m}^{p\top}\mathbf{x}_{n,m}. \tag{3.10}$$

In den letzten Komponenten des geschätzten Prädiktors befinden sich die Werte des Absolutwertes.

Es existieren also insgesamt drei unterschiedliche Modelle: das Modell der linearen Vorhersage ohne Absolutwert, das Modell der linearen Vorhersage mit Absolutwerten, wobei jedes Element des Absolutwert-Vektors identisch ist, und das Modell der linearen Vorhersage mit unabhängigen Komponenten im Absolutwert. Das erste Modell besitzt den Vorteil, dass weniger Vektoren zur Bestimmung eines Prädiktors benötigt werden. Da die Länge des Prädiktors eine untere Grenze der benötigten Beispiele zur Bestimmung desselbigen bietet, erhöht sich die Anzahl der benötigten Vektoren bei den anderen beiden Fällen. Im Fall, dass alle Komponenten des Absolutwertes gleich sind, bedeutet dieses, dass im Vergleich zu dem Fall ohne Absolutwert nur ein Vektor hinzu genommen werden muss. Im Fall, dass die Komponenten unabhängig sind, müssen so viele Vektoren hinzu genommen werden, wie die Dimension der Messwerte sind. Ferner sind dieses untere Grenzen, sodass je nach Variation der Trainingsdaten durchaus auch mehr Datenvektoren genutzt werden müssen.

3.2 Ereignisdetektion mittels eines Mischmodell-Ansatzes aus mehreren Prädiktoren

Im letzten Abschnitt wurde die Bestimmung von Gewichtsvektoren zur linearen Vorhersage diskutiert. Die lineare Vorhersage soll nun als Methode dienen, um Ereignisse zu detektieren. Hierfür sind drei Punkte notwendig. Diese sind die Bestimmung eines Modells, das Training des Modells und die Anwendung des trainierten Modells auf neue Messwerte.

Das hier vorgestellte Modell beruht auf den im letzten Abschnitt vorgestellten linearen Prädiktoren. Das heißt, auf Basis einer Trainingsmenge werden lineare Prädiktoren bestimmt. Da die Annahme, dass der Prozess, der die Daten generiert, global stationär ist, oft zu restriktiv ist, wird hier eine Abschwächung vorgenommen. Angenommen wird in diesem Abschnitt und in den entsprechenden Experimenten, dass der Prozess nur stückweise stationär ist. Das heißt, ein linearer Prädiktor soll über einen kurzen Zeitraum die Daten ausreichend beschreiben, allerdings kann ein anderer Abschnitt durch einen anderen Prädiktor beschrieben werden.

Die Größe, die für die Ereignisdetektion verwendet wird, ist der quadratische Prädiktionsfehler $\epsilon^2(n)$. Tritt der Normalfall ein, ist für zumindest einen linearen Prädiktor dieser Wert klein. Ist der quadratische Prädiktionsfehler groß für alle linearen Prädiktoren, so

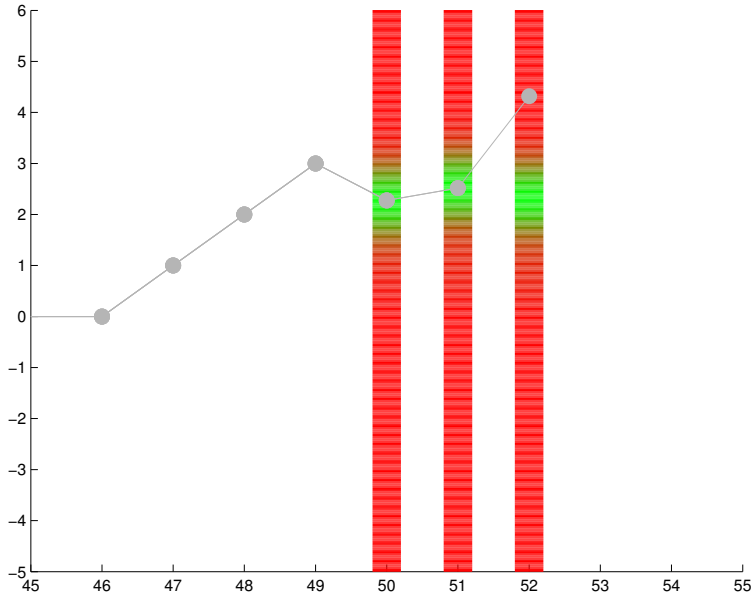


Abbildung 3.1: Schematische Darstellung der Ereignisdetektion mittels des Mischmodells. In den grünen Bereichen wird eine Messung als dem Normalfall angehörig bewertet, im roten Bereich als Ereignis. In zwei der drei markierten Bereiche ist der Normalfall beobachtet worden, im dritten bewerteten Messwert das Ereignis. Die ersten Messungen werden zur Beurteilung der nachfolgenden Messungen verwendet.

wird angenommen, dass ein Ereignis detektiert wurde. Da nur zwischen Ereignissen und dem Normalfall unterschieden wird (und nicht die Unterkategorie des Normalfalls, die durch einen spezifischen Prädiktor beschrieben wird, gesucht wird), wird eine Kombination dieser quadratischen Prädiktionsfehler verwendet.

Das Modell ist wie folgt: sei K die Anzahl der verwendeten linearen Prädiktoren, $\mathbf{w}_i$, mit $i = 1, 2, \ldots, K$. Sei ferner $\epsilon_i(n)^2$ der quadratische Prädiktionsfehler, mit dem eine lineare Vorhersage mittels des linearen Prädiktors $\mathbf{w}_i$ vom tatsächlich gemessenen Wert abweicht. Dann ist das Modell definiert durch

$$LPM(n) = \sum_{i=1}^{K} a(i) \exp\left(-\epsilon_i^2(n)\right), \tag{3.11}$$

wobei $a(i)$ Gewichte sind, das heißt $a(i) \geq 0$, $\sum_{i=1}^{K} a(i) = 1$. Ist $LPM(n)$ größer als ein zuvor festgelegter Schwellwert, so ist der Normalfall eingetreten; ist dieser kleiner, so wird angenommen, dass ein Ereignis eingetreten ist. Eine graphische Interpretation befindet sich in Abbildung 3.1.

3.2.1 Training eines Mischmodells

Das Training eines Mischmodells mit mehreren linearen Prädiktoren kann mittels einer Optimierung stattfinden. Hierbei seien die momentan geschätzten Prädiktoren durch $\mathbf{w}_i$ notiert, wobei der Index i die unterschiedlichen Prädiktoren nennt. Eine Unterscheidung zwischen geschätzten und wahren Prädiktoren wird in diesem Abschnitt nicht getroffen, da die wahren Prädiktoren unbekannt sind; somit sind alle in diesem Abschnitt besprochenen Prädiktoren geschätzt. Betrachtet wird zunächst eine einzelne Vorhersage mittels der geschätzten Gewichtsvektoren. Diese werden in dem Mischmodell nach Definition (3.11) kombiniert. Dadurch ergibt sich die Zielfunktion $LPM(n, \mathbf{W})$ mit den K geschätzten Prädiktoren

$$\begin{aligned} LPM(n, \mathbf{W}) &= \sum_{i=1}^{K} a(i) \exp\left(-\hat{\epsilon}_i^2(\mathbf{w})\right) \\ &= \sum_{i=1}^{K} a(i) \exp\left(-(\mathbf{X}^p(n)\mathbf{w}_i - \mathbf{x}(n))^\top (\mathbf{X}^p(n)\mathbf{w}_i - \mathbf{x}(n))\right), \end{aligned}$$

wobei $\mathbf{W} = \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_K\}$ die Menge der Prädiktoren ist. Der Gradient nach einem der Gewichtsvektoren $\mathbf{w}_i$ ist demnach im Fall der Prädiktion ohne Absolutglied

$$\begin{aligned}\nabla_{\mathbf{w}_i} LPM(n, \mathbf{W}) &= -2a(i) \cdot \exp\left(-\epsilon_i^2(n)\right) \\ &\quad \times \left(\mathbf{X}^{p\top}(n)\mathbf{X}^p(n)\mathbf{w}_i - \mathbf{X}^{p\top}(n)\mathbf{x}(n)\right) \\ &= 2a(i) \cdot \exp\left(-\epsilon_i^2(\mathbf{w})\right) \\ &\quad \times \mathbf{X}^{p\top}(n)\left(\mathbf{x}(n) - \mathbf{X}^p(n)\mathbf{w}_i\right).\end{aligned}$$

Dieses Ergebnis lässt sich für ein Optimierungsverfahren zum Training des Mischmodells folgendermaßen interpretieren. Einerseits bedeutet der Faktor $2a(i) \cdot \exp\left(-\epsilon_i(n)^2\right)$, dass Länge des Gradienten $\|LPM(n, \mathbf{W})\|_2$ die Präzision der Vorhersage berücksichtigt. Prädiktoren, die gute Vorhersagen liefern, werden stärker beeinflusst als Prädiktoren, deren Vorhersage eventuell für andere Segmente genauer zutrifft. Andererseits ist die Länge des Gradienten auch davon abhängig, wie präzise die Vorhersage mittels der aktuellen Prädiktoren an sich getätigt werden kann: je genauer die Vorhersage, desto kleiner ist die Länge der Differenz zwischen dem wahren Wert $\mathbf{x}(n)$ und der Vorhersage $\hat{\mathbf{x}} = \mathbf{X}^p(n)\mathbf{w}_i$, also $\|\mathbf{x}(n) - \mathbf{X}^p(n)\mathbf{w}_i\|_2$. Insgesamt werden vor allem die Prädiktoren, die zwar akzeptable, wenn auch nicht präzise Vorhersagen in dem betrachteten Segment tätigen, durch einen Optimierungsschritt nach dem Gradientenverfahren am meisten beeinträchtigt. Um eine Überanpassung zu vermeiden, sollte entsprechend die Schrittweite der gesamten Prognose gesetzt werden. Hierfür bietet sich der Faktor $c(n) = \left(\sum_j \exp(-\epsilon_j^2(n))\right)^{-1}$ an. Dadurch werden Segmente, in denen besonders gute Vorhersagen getroffen werden, weniger stark berücksichtigt (und dadurch die entsprechenden Prädiktoren nur wenig verändert) als solche, die noch nicht durch einen Prädiktor repräsentiert werden. Dadurch ist nach Abschluss des Trainings sichergestellt, dass die finalen Prädiktoren die gegebene Trainingsmenge repräsentieren können.

Der Gewichtsvektor $\mathbf{a}$ kann theoretisch ebenfalls mit trainiert werden. Allerdings beschränken sie in diesem Fall auch die Aktualisierungen der Prädiktoren innerhalb der Optimierungsschritte: ein kleines Gewicht führt zu einer langsamen Adaption. Daher ist es ratsam, die Gewichte für das Training der Prädiktoren auf $a(i) = K^{-1}$ zu setzen und sie anschließend zu bestimmen. Die Regel zum Aktualisieren eines Gewichtsvektors lautet also

$$\hat{\mathbf{w}}_i \leftarrow \frac{2c(n)\exp\left(-\epsilon_i^2(\mathbf{w})\right)}{K} \cdot \mathbf{X}^{p\top}(n)\left(\mathbf{x}(n) - \mathbf{X}^p(n)\mathbf{w}_i\right). \tag{3.12}$$

Dieses entspricht einem statistischen Gradientenverfahren zum Training des Mischmodells. Somit lässt sich ein praktisch anwendbares Verfahren zum Training dieses Modells finden.

Allerdings ist dieses nicht notwendigerweise schnell im Sinne der benötigten Rechenoperationen, da immer mehrere Schritte für eine Optimierung benötigt werden. Ein Optimierungsverfahren höherer Ordnung ist ebenfalls möglich und würde das Training gemäß der Theorie beschleunigen [8], zum Beispiel mittels eines Quasi-Newton-Verfahrens wie BFGS [21]. Allerdings existiert noch eine Möglichkeit, die Prädiktoren in einem einzigen Schritt zu berechnen. Da, wie im letzten Abschnitt erläutert, ein linearer Prädiktor direkt für Segmente der Trainingsdaten bestimmt werden kann, ist dieses Verfahren vorzuziehen; hierdurch wird nur ein Bruchteil der benötigten Rechenzeit, im Vergleich zum statistischen Gradientenverfahren, verwendet. Die Idee dabei ist, dass Prädiktoren nur für Segmente der Trainingssequenz geschätzt werden, die mit vorher geschätzten Prädiktoren schlecht repräsentiert werden. Dadurch erhält man eine Menge von K Prädiktoren, die den Normalfall insgesamt repräsentieren können. Zudem erhält man die Möglichkeit, die Zahl der Prädiktoren anzupassen.

Da vor dem Training kein Prädiktor zugänglich ist und somit alle Daten gleich repräsentiert werden, kann man zur Initialisierung des Trainings zunächst $\mathbf{X}_{n_1,m_1}$ zufällig auswählen. Hierfür sei die Anzahl der für einen einzelnen Prädiktor verwendeten Datenwerte L konstant, sodass stets $m_1 = n_1 + L$ gilt. Der Index n_1 ist der zufällig gewählte Wert. Ist das Segment gewählt, so wird nach dem gewählten Modell der Prädiktor $\mathbf{w}_1$ bestimmt, wie in Abschnitt 3.1 beschrieben. Anschließend wird die Güte des Prädiktors errechnet. Hierfür wird für jedes mögliche n der Wert $e_1(n) = \exp(-\epsilon_1^2(n))$ genommen. Nach Definition ist der Prädiktor $\mathbf{w}_1$ gut geeignet, falls $e_1(n)$ hoch ist und für ein Segment ungeeignet, falls dieser Wert klein ist. Somit wird der nächste Index gegeben durch $n_2 = \arg\min_n e_1(n)$ und entsprechend der zweite Prädiktor $\mathbf{w}_2$ bestimmt. Ein Segment ohne repräsentativen Prädiktor hat sowohl für den ersten als auch zweiten Prädiktor einen hohen Prädiktionsfehler. Daher ist $e_2(n) = e_1(n) + \exp(-\epsilon_2^2(n))$ und $n_3 = \arg\min_n e_2(n)$. Sukzessive werden somit K Prädiktoren erzeugt, die den Normalfall beschreiben. Das Verfahren zeigt in Experimenten, die in Rahmen dieser Arbeit durchgeführt wurden, eine deutlich höhere Geschwindigkeit als Verfahren, die auf dem Gradienten basieren, bei gleichbleibender Erkennungsrate der Ereignisse.

Dieses Trainingsverfahren lässt sich sehr einfach auf die anderen beiden Modelle mit Absolutglied übertragen. Hierfür wird die Matrix $\mathbf{X}^p_{n,m}$ durch die Matrix $\mathbf{X}'^p_{n,m}$

respektive $\mathbf{Y}^p_{n,m}$ ersetzt, wie im vorherigen Abschnitt beschrieben. Die Bestimmung der Prädiktoren bleibt äquivalent erhalten.

Sowohl bei dem Gradientenverfahren als auch bei der sukzessiven Bestimmung von Prädiktoren erhält man eine Menge von K Prädiktoren. Allerdings existieren für natürliche Daten oft Prädiktoren, die sehr gute Vorhersagen für sehr viele Segmente liefern, während andere bei sehr speziellen Ausprägungen eine präzise Vorhersage machen. Das Mischmodell soll diesem Verhalten genügen, demzufolge die Prädiktoren hoch gewichten, die oft für eine Vorhersage notwendig sind, und entsprechend kleinere Gewichte $a(i)$ für Prädiktoren beinhalten, die als Korrektiv diese allgemeinen Prädiktoren unterstützen. Hierfür sei $b_i = \sum_n \exp(-\epsilon_i^2(n))$. Dann ist das Gewicht $a(i)$ durch $a(i) = \frac{b_i}{\sum_j b_j}$ gegeben. Dieses gilt für beide Trainingsverfahren gleichermaßen. Dadurch ist die Bedingung für die Gewichte erfüllt, das heißt, die Summe über alle Gewichte ist eins und jedes Gewicht ist nichtnegativ.

Tests legen nahe, dass beide Trainingsverfahren vergleichbare Ergebnisse liefern. Dieses gilt für alle in diesem Kapitel beschriebenen linearen Vorhersagemodelle. Da die sukzessive Bestimmung von Prädiktoren deutlich schneller als die Verwendung gradientenbasierter Methoden ist, weil hier sehr einfach und schnell die benötigten Prädiktoren bestimmt werden können, wurde dieses Verfahren für die Experimente in Kapitel 3.3 verwendet.

3.2.2 Anwendung eines trainierten Mischmodells

Im letzten Abschnitt wurde das Training eines Mischmodells diskutiert, das auf linearen Prädiktoren beruht. Dieses Training wird anhand einer Basis von Datenvektoren vorgenommen, die den Normalfall definieren. Das trainierte Modell soll nun angewendet werden, um neue Datenvektoren als Normalfall beziehungsweise als Ereignis zu klassifizieren. Eine graphische Interpretation befindet sich in Abbildung 3.2.

Wie im letzten Abschnitt beschrieben, existiert ein Maß $LPM(n, \mathbf{W})$, das mittels der Prädiktoren $\mathbf{W}$ einen Wert dafür gibt, wie gut ein Segment, indiziert mit n, zu dem Normalfall passt. Nach Definition gilt für alle n die Eigenschaft $0 \leq LPM(n, \mathbf{W}) \leq 1$, wobei ein geringer Wert für eine Abweichung vom Normalfall steht. Dadurch ist es möglich, durch einen Schwellwert den Normalfall vom Ereignis abzugrenzen.

Dieser Schwellwert lässt sich aus den gegebenen Trainingsdaten bestimmen. Allerdings kann ein Schwellwert, der garantiert, dass alle Trainingsdaten als Normalfall klassifiziert werden, zu konservativ sein. Daher ist ein höherer Schwellwert, der zum Beispiel fünf Prozent der Trainingsdaten fälschlicherweise als Ereignis klassifiziert, in

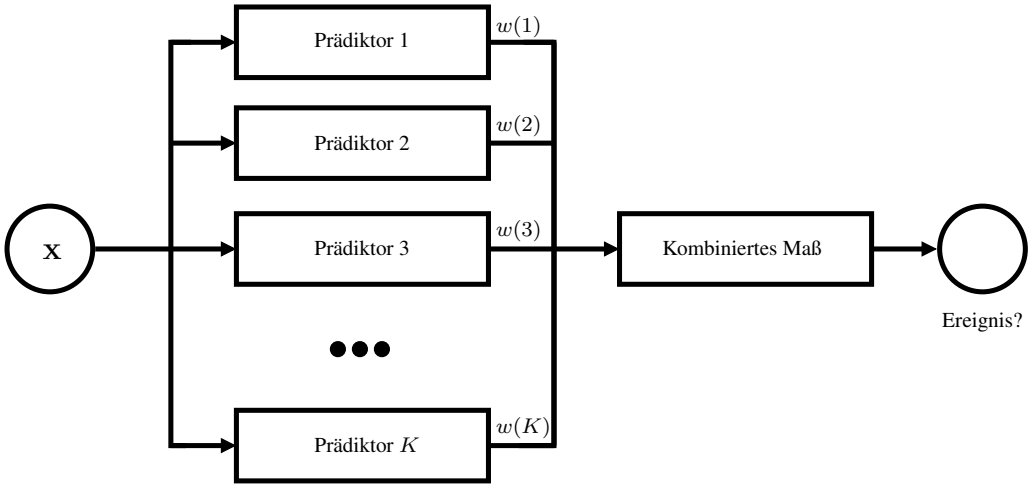


Abbildung 3.2: Grundsätzlicher Aufbau der Ereignisdetektion mittels eines linearen Mischmodells. Die Messwerte (hier repräsentiert als $\mathbf{X}$) werden individuell durch die Prädiktoren beurteilt. Deren Ergebnisse gehen gewichtet in ein gemeinsames Maß ein, mit dem entschieden wird, ob ein Ereignis stattgefunden hat oder nicht.

der Regel vorzuziehen. Als Korrektiv kann dann $LPM(n, \mathbf{W})$ über den Zeitindex n mittels eines Tiefpasses gefiltert werden, da wahre Ereignisse in praktischen Problemen oft nicht nur isoliert in einem einzelnen Messwert, sondern in mehreren sukzessiven Messwerten beobachtet werden.

3.3 Experimente bezüglich des Mischmodellansatzes

Der Test für das Mischmodell auf Basis von linearen Prädiktoren besteht aus zwei Abschnitten. Auf synthetischen Daten wird zunächst das Mischmodell mit einem GMM und einem HMM verglichen. Anschließend wird das Mischmodell auf ein Testszenario verwendet, bei dem es um die Analyse von Videosequenzen geht.

In den Videosequenzen wird ein Modellfahrzeug verfolgt. Der Normalfall ist, wenn das Fahrzeug sich normal im sichtbaren Bereich bewegt. Dabei sind alle Bewegungen erlaubt, die das Fahrzeug durchführen kann, mit Ausnahme von Rückwärtsbewegungen. Das Ereignis ist das Treffen eines Objektes.

Der Einfachheit halber ist der Hintergrund unbewegt gehalten, sodass das zu beobachtende Objekt leicht zu verfolgen ist. Dadurch genügt ein einfacher Algorithmus zur Trennung von Vorder- und Hintergrund, um das zu beobachtende Objekt zu verfolgen.

Dabei wird eine Kodierung des Bildes verwendet: Ein Pixel ist 1, wenn das entspre-

chende Pixel des Bildes der Videosequenz zum Vordergrund gehört, und 0 sonst [62]. Aus dem Vordergrund werden anschließend die Merkmale extrahiert. Dies ist notwendig, da die drei in diesem Experiment verglichenen Modelle nicht auf der Information, welche Pixel zum Vordergrund gehören, arbeiten, sondern mit stetigen Messwerten. Die Modelle werden anschließend auf Merkmale trainiert, die aus Videosequenzen extrahiert wurden, die nur dem Normalfall entsprechen. Anschließend werden die trainierten Modelle auf Sequenzen angewendet, die ebenfalls das Ereignis enthalten können. Die Aufgabe der Modelle ist ausschließlich, das Ereignis zu detektieren.

3.3.1 Synthetische Daten

Für den Vergleich des Modells werden Merkmale verwendet, die durch ein HMM erster Ordnung generiert wurden, dessen Übergangswahrscheinlichkeiten und sonstigen Parameter manuell festgelegt werden und nur vom vorherigen Zustand abhängen. Der Grund für den Vergleich auf synthetischen Daten ist, dass hierbei das Ereignis zu einem genau festgesetzten Zeitpunkt mit einer kontrollierten Modifikation eingesetzt werden kann. Dadurch bieten diese Daten genaue Informationen über das Ereignis, was durch die Verwendung von realen Daten abhängig ist von einer vorherigen Segmentierung, die aufwendig und fehleranfällig sein kann. Die synthetischen Daten sind für einen direkten Vergleich der Modelle daher besser geeignet.

Da eines der Modelle, die verglichen werden, ein HMM ist, wird kein standardmäßiges Modell zum Erstellen der synthetischen Daten verwendet, damit die Daten nicht einem der theoretischen Modell exakt entsprechen und somit die Daten realistischer sind. Stattdessen wird jedem Zustand ein anderer linearer Filter zugeordnet. Das Testsignal ist demnach stückweise stationär und entspricht den Annahmen des Mischmodells auf Basis der linearen Vorhersage. Gleichzeitig können auch das GMM und das HMM auf diese Daten trainiert werden.

Das Testsignal wird erzeugt mittels

$$\mathbf{x}(n) = \mathbf{m}_s + \overline{\mathbf{x}}_s + \mathbf{v}(n) \tag{3.13}$$

mit

$$\overline{\mathbf{x}}_s = \sum_{i=1}^{3} a_s(i)(\mathbf{x}(n-i) - \hat{\mu}(n)),$$
$$\hat{\mu}(n) = \frac{1}{3}\sum_{i=1}^{3} \mathbf{x}(n-i),$$
$$\mathbf{v}(n) \sim N(\mathbf{0}, \Sigma_s).$$

Hierbei sind $\mathbf{m}_s$, $a_s(1)$, $a_s(2)$, $a_s(3)$ sowie Σ_s Parameter des Zustands s des HMMs, das zur Simulation genutzt wird.

Der Vorteil dieses Modells ist, dass die Daten stückweise stationär sind, das heißt, dass sie sich in sehr kurzen Zeitsegmenten linear beschreiben lassen, global hingegen wie ein HMM unterschiedliche Zustände aufweisen und dadurch nicht stationär sind. Diese Verbindung ergibt in der Simulation realistischere Daten als ein HMM oder ein üblicher linearer Filter, welche die üblichen Methoden in diesem Fall der Simulation sind. Ferner entspricht keines der Modelle, die in diesem Experiment verglichen werden, dem Modell, das zur Merkmalserzeugung genutzt wird. Dadurch bietet dieser Test die nötigen Ansatzpunkte zur Analyse der Verfahren.

Das generierte Signal ist fünfdimensional, und das HMM besitzt fünf Zustände. Es wurden unterschiedliche Testmengen für die Tests generiert. Sowohl für das Ereignis als auch den Normalfall wurden 50000 Datenwerte generiert. Für das Ereignis werden die Werte $\mathbf{m}_s$ verändert (Satz 1), die Filterkoeffizienten wurden modifiziert (Satz 2 und Satz 3) sowie normalverteiltes Rauschen genommen (Satz 4).

Alle drei Modelle verwenden Schwellwerte, um zu kontrollieren, ab wann ein Messwert zu dem Ereignis gerechnet werden kann: ist im Falle eines HMMs oder GMMs die Wahrscheinlichkeit oder im Fall des Mischmodells das vorgestellte Maß geringer als ein zuvor festgelegter Schwellwert, wird angenommen, das Ereignis beobachtet zu haben. Diese Eigenschaft ermöglicht es, diese drei Methoden anhand von Grenzwertoptimierungskurven (engl. *receiver operating characteristic*, ROC [5]) direkt zu vergleichen.

3.3.2 Extraktion des Vordergrundes und Bestimmung der Merkmale

Zur Bestimmung des Vordergrundes innerhalb der Experimente, die auf Videosequenzen beruhen, wird zunächst ein Modell des Hintergrundes erstellt. Unterscheidet sich ein

Pixel eines neuen Bildes in der Videosequenz von diesem Modell, so wird dieser dem Vordergrund zugeordnet [62].

Die Testvideos wurde mit fünfzehn Bildern pro Sekunde aufgenommen. Für die Entscheidung des Hintergrundes werden Intensitätsbilder verwendet, das heißt $\mathbf{F}_n$ mit $F_n(i,j) \in [0,1]$. Die ersten drei Sekunden wurden dafür verwendet, das Hintergrundmodell zu initialisieren; das heißt, das Hintergrundmodell $\mathbf{B}_0$ ist der Mittelwert der ersten fünfzehn Bilder der Aufnahme. Sei der Einfachheit halber $\mathbf{F}_1$ das erste Bild, das nicht zur Initiierung des Hintergrundes verwendet wird. Ferner sei $\mathbf{T}_0$ die Initiierung eines Schwellwertes für jeden Pixel mit $T_0(i,j) = 0.5$. Dieser Schwellwert wird ebenso mit Hilfe neuer Bilder angepasst wie das Hintergrundmodell. Die Verwendung eines Schwellwertes für jeden Pixel erlaubt für jeden Pixel ein unterschiedliches Rauschen.

Ein Pixel mit dem Index (i,j) des Bildes $\mathbf{B}_n$ gehört genau dann zum Vordergrund, wenn gilt $\|C_n(i,j) - B_n(i,j)\|_2^2 > T_n(i,j)$. Daher sei $\mathbf{I}_n$ mit

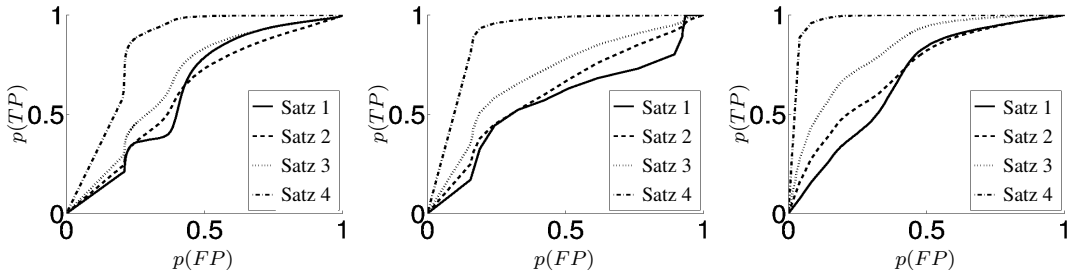
$$\begin{aligned} D_n(i,j) &= \|C_n(i,j) - B_n(i,j)\|_2^2, \\ I_n(i,j) &= [\![D_n(i,j) > T_n(i,j)]\!] \end{aligned} \tag{3.14}$$

das Bild dieses Vordergrundes. Sei ferner $\alpha_B \in (0,1)$ die Anpassungsrate an ein neues Bild. Dann wird der Hintergrund respektive der Schwellwert aktualisiert mit

$$\begin{aligned} B_{n+1}(i,j) &= (1 - I_n(i,j)) \cdot (\alpha_B B_n(i,j) + (1-\alpha_B) C_n(i,j)) + I_n(i,j) \cdot B_n(i,j), \\ T_{n+1}(i,j) &= (1 - I_n(i,j)) \cdot (\alpha_B (T_n(i,j) + 0.01) \\ &\qquad + (1-\alpha_B) D_n(i,j)) + I_n(i,j) \cdot T_n(i,j). \end{aligned}$$

Mit diesem Algorithmus wird der Vordergrund vom Hintergrund getrennt. Das Ergebnis ist eine Sequenz von Vordergrundbildern $\mathbf{I}_n$. Die Position des beobachteten Objekts sei definiert als der geometrische Schwerpunkt aller Pixel, die in einem Bild zum Vordergrund gehören.

Die Ereignisdetektion geschieht in diesem Experiment über die Bewegungen des Objekts. Da die Richtungen für die Bewegung uninteressant sind, werden die Sequenzen vor der Prädiktion gedreht, sodass der Differenzvektor zwischen den frühesten beiden Positionen stets in dieselbe Richtung zeigen. Auf diese Art wird die Anzahl der benötigten linearen Prädiktoren reduziert, das Prinzip des Modells wird dadurch nicht beeinflusst.



(a) GMM (b) HMM (c) Mischmodell

Abbildung 3.3: Grenzwertoptimierungskurven (ROCs) für die drei verglichenen Modelle anhand der unterschiedlichen Ereignissimulationen. Die Testsätze wurden wie in Abschnitt 3.3.1 erläutert erzeugt. Hierbei wird in Abhängigkeit eines Schwellwerts eine Detektionswahrscheinlichkeit $p(TP)$ gegen die Wahrscheinlichkeit eines fälschlicherweise als Ereignis klassifizierter Normalfall $p(FP)$ aufgetragen. In (a) sind die Ergebnisse für ein GMM dargestellt. In (b) sind zum Vergleich die ROCs für ein HMM abgebildet. In (c) sind die Ergebnisse des Mischmodellansatzes auf Basis der linearen Prädiktoren zu sehen.

3.3.3 Ergebnisse

Im Folgenden werden die Ergebnisse der Experimente diskutiert. Im ersten Abschnitt wird insbesondere das Mischmodell auf Basis von linearen Prädiktoren mit HMMs und GMMs verglichen. Dadurch wird gezeigt, dass Fälle existieren, bei denen dieses Modell eingesetzt werden kann. Im anschließenden Test wird anhand eines praktischen Beispiels eine Anwendung gezeigt.

3.3.3.1 Ergebnisse des Vergleichs zwischen HMMs, GMMs und dem Mischmodell auf Basis linearer Prädiktoren

In Abbildung 3.3 sind die Grenzwertoptimierungskurven für die drei zu vergleichenden Modelle dargestellt, das sind das GMM, das HMM und das Mischmodell auf Basis von linearen Prädiktoren. Die Anzahl der Zustände im Markov-Modell des HMM sowie die der Normalverteilungen im GMM sowie jedem Zustand des HMM wurden derart gewählt, dass für diese Testmengen die bestmöglichen Ergebnisse dargestellt werden konnten; das heißt, die Fläche unterhalb der ROC-Kurven wurde bezüglich der Parameter der Modelle optimiert. Dadurch ist der bestmögliche Vergleich dieser Modelle mit dem Mischmodell auf Basis der linearen Prädiktoren möglich. Dabei werden für das GMM eine Mischung von zehn normalverteilten Zufallsvariablen angenommen,

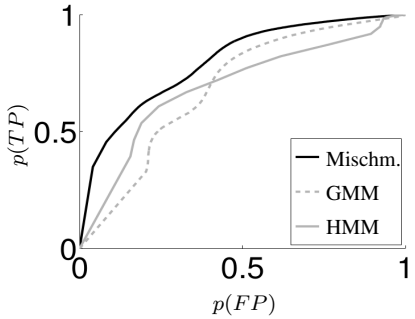


Abbildung 3.4: Vergleich des GMMs, HMMs und des Mischmodells über alle vier Testklassen.

ebenfalls besitzt das HMM zehn Zustände. In jedem Zustand wird die Beobachtung als normalverteilt angenommen. Das Mischmodell basiert auf fünfzig linearen Prädiktoren, die jeweils eine Vorhersage aus zehn Beobachtungen schätzen.

In Abbildung 3.4 ist ein direkter Vergleich der ROC-Kurven gemittelt über alle vier Testsätze dargestellt. Dieser Vergleich besitzt insgesamt die beste Generalisierung, um die Möglichkeiten der Modelle einzuschätzen. Wie man in dieser Darstellung sieht, ist insgesamt das vorgestellte Mischmodell auf Basis der linearen Prädiktoren der beste der drei Klassifikatoren für die gegebenen Daten. Die Aussage dieses Experiments ist, dass Probleme existieren können, die mit diesem Modell besser behandelt werden können als mit den anderen beiden getesteten Klassifikatoren. Daher ist es ein möglicher Kandidat für ein System zur Ereignisdetektion.

3.3.3.2 Ergebnisse des Tests der Analyse von Bewegungsdaten

In diesem Experiment wurde nur das Mischmodell auf Basis der linearen Prädiktoren verwendet. Das Experiment dient dazu, zu zeigen, dass es in praktischen Anwendungen verwendet werden kann. Es bietet somit das Gegenstück zu dem vorherigen Experiment: Während dort das Ziel war, die Qualität der Verfahren zu beurteilen, ist es hier die Vorstellung eines konkreten Beispiels. Da der genaue Zeitpunkt des Ereignisses, das heißt das Treffen des Objekts auf ein Hindernis, nicht manuell eindeutig zugeordnet werden kann, ist ein direkter Vergleich zwischen den drei Modellen in diesem Experiment schwierig. Daher wurde sich dafür entschieden, den Vergleich in diese beiden Teile, einen mit Hilfe synthetischer Daten und damit eindeutig zutreffenden Ereignissen und einen praktischen Anteil, in dem das Mischmodell anhand reeller Daten verwendet wird, zu teilen.

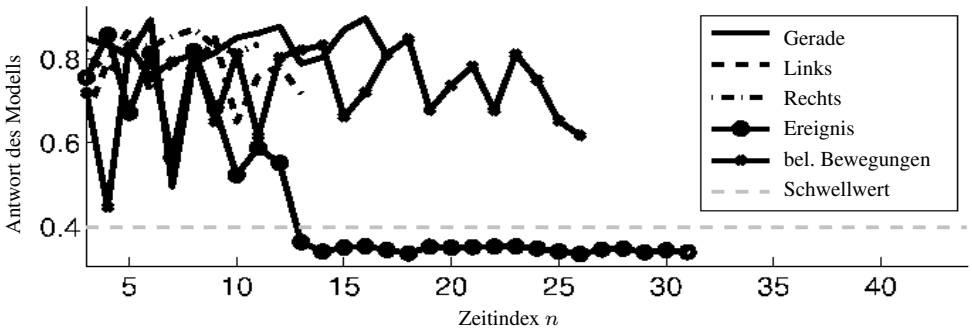


Abbildung 3.5: Antwort des Modells auf Beispielvideos, die unterschiedliche Bewegungen des verfolgten Objekts darstellen (skaliert). Je geringer dieser Wert, desto wahrscheinlicher ist die Beobachtung eines Ereignisses. Das wahre Ereignis ist eindeutig von den normalen Bewegungen zu unterscheiden.

Hierbei führt das ferngesteuerte Fahrzeug unterschiedliche, teilweise komplexe Bewegungen aus. Auf diese Bewegungen wird das Modell trainiert. Es werden nur drei Prädiktoren verwendet, die jeweils aus drei Beobachtungen eine neue Beobachtung schätzen. Diese geringere Anzahl an zu schätzenden Parametern hat sich als hinreichend erwiesen [52].

In Abbildung 3.5 ist die Antwort des Modells angegeben, sowie ein möglicher Schwellwert für das Problem, die Ereignisse in diesem Signal zu detektieren. Hierbei ist die Antwort des Modells gegen den Zeitindex aufgetragen. Die unterschiedlichen Bewegungen werden alle dem Normalfall zugeordnet. Ausschließlich das Ereignis ist als solches detektiert. Ferner ist zu sehen, dass die Bewegungen vor der Kollision, die offensichtlich dem Normalfall entsprechen, auch dem Normalfall zugeordnet werden. Hiermit ist ersichtlich, dass dieses Modell für dieses Problem geeignet ist und die Parameter hinreichend gewählt wurden. Dadurch ist gezeigt, dass dieses Verfahren zur Ereignisdetektion genutzt werden kann.

In Abbildung 3.6 sind einige Beispielbilder aus den Experimenten zu sehen. Hierbei wird das Ereignis gezeigt, also das Auftreffen des Fahrzeuges auf ein Hindernis. Dieses Ereignis wird anhand einer Markierung in dem oberen linken Bereich eines Bildes dargestellt. Ferner sind die detektierten Vordergrundpixel zu sehen. Das Objekt, mit dem das Fahrzeug kollidiert, ist nicht in der Detektion des Vordergrundes enthalten. Dieses wurde absichtlich so durchgeführt, dass ausschließlich die Bewegungen des Fahrzeuges zur Detektion genutzt werden. Wie in diesen Bildern ersichtlich, sind die Bewegungen

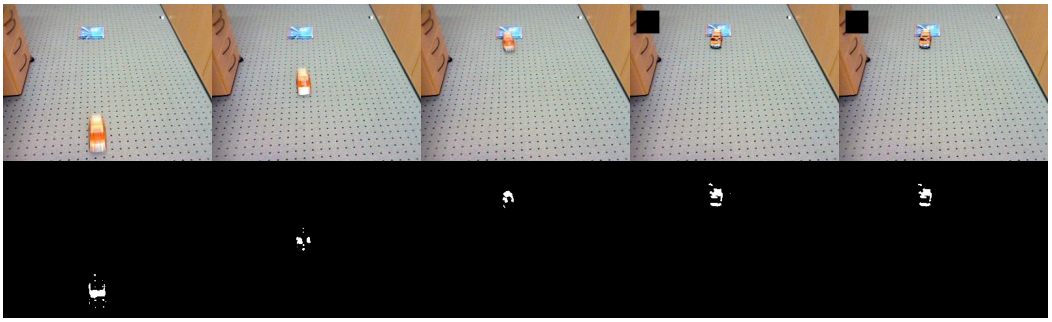


Abbildung 3.6: Beispielaufnahme des Experiments. Die Markierung im oberen linken Bereich des Bildes zeigt die Detektion eines Ereignisses an. Im unteren Bereich sind die gemessenen Vordergrundpixel bildlich dargestellt (weiß).

vor der Kollision als Normalfall klassifiziert worden. Das Ereignis wurde demnach auch zum richtigen Zeitpunkt klassifiziert.

3.3.4 Interpretation der Ergebnisse

Die Experimente haben gezeigt, dass das Mischmodell auf Basis der linearen Prädiktoren zur Ereignisdetektion geeignet ist. In den Versuchen mittels synthetischer Daten hat sich dieses Modell als effektiver als HMMs und GMMs erwiesen. Diese beiden Modelle gelten als Standardmodelle in der Ereignisdetektion [44].

In dem zweiten Experiment wurde das Mischmodell auf Basis der linearen Prädiktoren auf ein Problem angewendet, wie es zum Beispiel bei der Verkehrsüberwachung auftreten kann. Hierbei werden Fahrzeuge beobachtet, ungewöhnliche Ereignisse, wie in etwa Unfälle, sollen detektiert werden. Die Experimente haben gezeigt, dass das vorgestellte Modell auf diese Probleme anwendbar ist. Auf einen Vergleich mit GMMs und HMMs wurde in diesem Experiment aus zwei Gründen verzichtet. Erstens ist das Ereignis nicht einem einzelnen Bild zuzuordnen; ein Vergleich, welches Modell das Ereignis zum korrekten Zeitpunkt detektiert, ist somit schwierig. Zweitens hat sich in anfänglichen Versuchen gezeigt, dass diese Modelle nicht geeignet sind, ausschließlich mittels der Position die Ereignisse zu detektieren. Eine weitere Datentransformation hätte zusätzliche Informationen in diesen Modellen bedeutet, sodass die Vergleichbarkeit abgenommen hätte.

3.4 Diskussion des Mischmodell-Ansatzes

In diesem Kapitel wurde ein Mischmodell-Ansatz zur Ereignisdetektion vorgestellt. Dieses Modell kombiniert eine Menge von linearen Prädiktoren und nutzt den Prädiktionsfehler zur Bestimmung eines Ereignisses. Das Modell ist einfach zu trainieren und anzuwenden: die Prädiktoren werden in einem geschlossenen Modell verwendet, ein Ereignis wird detektiert, wenn die Prädiktoren nicht die neuen Messewerte vorhersagen können, wenn also ein großer Prädiktionsfehler entsteht.

Allerdings nimmt dieses Modell eine stückweise Stationarität des Signals an. Auch werden implizit einige Annahmen an die Verteilungen der Fehler gestellt, die oft nicht zutreffen. Gerade bei sehr komplexen Systemen ist dieses Modell zu restriktiv. In den nächsten Kapiteln wird ein Verfahren behandelt, das auf datengetriebenen Modellen beruht. Diese erweitern die Anwendungsmöglichkeiten der Ereignisdetektion um Verfahren, die auch in allgemeineren Situationen benutzt werden können.

4 Bedingte Zufallsfelder und Maximum-Entropy-Markov-Modelle

Markov-Modelle werden oft zur Modellierung von zeitabhängigen Daten verwendet. Eines der am häufigsten verwendeten Modelle, die HMMs, wurde bereits besprochen. Dieses Modell wurde zum Beispiel in [65] für Sprachsignale angewendet. Die HMMs gehören zu der Gruppe der generativen Modelle, dadurch sind sie insbesondere zur Simulation von Merkmalen geeignet [41]. Hierbei wird eine Verteilung der Daten selbst modelliert, die demnach als bekannt angenommen werden muss. Bei der Simulation werden hieraus Beispiele gezogen.

In diesem Kapitel werden Markov-Modelle besprochen, die datengetrieben oder deskriptiv sind. Diese Modelle sind prinzipiell nicht zur Simulation geeignet. Dafür stellen sie deutlich weniger Annahmen an die Verteilung der Daten, sodass sie interessante Methoden speziell zur Analyse von Daten sind: Durch den Verzicht der Simulation werden unbekannte Verteilungen akzeptiert. Dadurch ist es nicht notwendig, Fehler in der Modellierung der Daten selbst zu akzeptieren. Diese Eigenschaft macht derartige Modelle für die Ereignisdetektion interessant, da hier oft das Modell für den Normalfall ausschließlich aus Messwerten bestimmt wird, somit prinzipbedingt die tatsächliche Verteilung unbekannt ist. Ferner wird auch die Verwendung von inhomogenen Sensornetzwerken vereinfacht, bei denen die Verteilung beliebig kompliziert sein kann, zum Beispiel können Sensoren mit Zählgrößen und stetigen Größen in einem Netzwerk vorhanden sein.

Die Bestimmung der Verteilung des Normalfalls ist bei der Ereignisdetektion von besonderer Bedeutung. Hierin befinden sich alle Informationen, die genutzt werden, um das Ereignis zu detektieren; ebenfalls beeinflussen Annahmen das Ergebnis einer Entscheidung. Demnach sollte bei der Modellierung des Normalfalls besondere Sorgfalt angewendet werden, die vorhandenen Informationen möglichst vollständig genutzt und Annahmen so weit wie möglich vermieden werden. Ein Prinzip, nach dem man hierbei vorgehen kann, ist das Prinzip der maximalen Entropie, das später erläutert wird und die Grundlage der hier verwendeten Methoden bildet. Dieses Prinzip wird in Abschnitt 4.2

besprochen. Bei CRFs und MEMMs wird es auf die Zustandssequenzen angewendet; die Merkmale dienen dazu, die Übergänge zu bestimmen und beeinflussen die Verteilungen der Zustände. Dieser Zusammenhang wird in Abschnitt 4.1 thematisiert.

CRFs und MEMMs sind miteinander verwandt und können gemeinsam in einer geschlossenen Theorie betrachtet werden. Sie sind zunächst nicht für die Ereignisdetektion vorgestellt worden und besitzen Eigenschaften, die die direkte Anwendung zur Ereignisdetektion verhindern. In den Abschnitten 4.3 und 4.4 werden diese Modelle zunächst in der allgemeinen Form [41, 54] besprochen. Ferner wird eine neue Methode vorgestellt, die Vorteile beider Modelle zu vereinen. Diese Methode, die im Rahmen dieser Doktorarbeit entwickelt wurde, ist für die Ereignisdetektion sehr interessant, da die vereinigten Eigenschaften die meisten Vorteile bieten. Zuletzt werden übliche Verfahren zum Training der Modelle diskutiert. Die Anpassungen zur Ereignisdetektion werden in Kapitel 5 besprochen.

4.1 Generative und deskriptive Modelle

CRFs und MEMMs gehören zu den deskriptiven Modellen [41, 54]. Diese Modelle basieren darauf, dass sie von Merkmalen abhängig sind und somit eine Interpretation der Daten selbst darstellen, im Gegensatz zu generativen Modellen, bei denen die Datenerzeugung modelliert wird.

CRFs und MEMMs werden oft anhand von Beispieldaten trainiert. Es wird dabei das Modell derart angepasst, dass es auf die Daten in der Form reagiert, wie anhand bekannter Beispiele bekannt ist: Nimmt bei einer bestimmten Messung und bei bestimmten Nachbarn in der Zustandssequenz ein Zustand in diesen Beispielen immer die Farbe ζ_k an, so soll das Modell diesem Zustand dieselbe Farbe zuordnen, wenn in einer neuen Sequenz dieselbe Situation auftritt. Für das Training werden für gewöhnlich zwei Informationen benötigt: Beispieldaten und die erwünschten Antworten des Modells in Form von gefärbten Zustandssequenzen. Eine Erweiterung des Trainings eines deskriptiven Modells, bei dem die Antworten des Modells nicht benötigt werden, wird in Abschnitt 4.7.2 thematisiert. Dieses Training ist eine Erweiterung, die im Zusammenhang dieser Arbeit erstellt wurde, sie ist somit nicht Teil der ursprünglichen Beschreibung von CRFs und MEMMs.

HMMs können ebenfalls anhand der Information von Messwerten und einer Zustandssequenz trainiert werden; üblich ist auch ein Training ohne die bekannte Zustandssequenz [65]. Im Training werden die Parameter des Modells derart angepasst, dass es Merkmale

erzeugen kann, die den Trainingsdaten möglichst ähnlich sind [5]. Ist der Prozess, mit dem die Daten generiert wurden, sehr unterschiedlich zu einem HMM, so wird das HMM unweigerlich keine vergleichbaren Daten liefern können. Dadurch ist die Aussage des Modells, neue Daten zu bewerten, ebenfalls eingeschränkt, da mit diesem Modell nicht die tatsächliche Verteilung der Merkmale bestimmt wird.

Üblich ist es für diese Bewertung, die Likelihood für Merkmale bei gegebenen Modell zu bestimmen und als Gütemaß zu verwenden. Ist die Verteilung des Modells nicht korrekt, entspricht auch die Likelihood nicht der Realität. Dadurch ist die Aussagekraft des Gesamtmodells hinsichtlich seiner Annahmen beschränkt.

Die im Folgenden besprochenen Modelle basieren auf dem Prinzip der maximalen Entropie. [54, 63]. Daher sind die einfacheren Modelle auch "Markov-Modelle der maximalen Entropie" (engl. *maximum entropy Markov model*) genannt worden [54]. Die darauf aufbauenden, komplexeren Modelle, bei denen ein ungerichtetes Markov-Zufallsfeld die beschreibende Eigenschaft ausführt, wurde "bedingtes Zufallsfeld" (engl. *conditional random fields*) genannt [41]. Bei dieser Namensgebung steht im Vordergrund, dass CRFs datengetrieben sind; dennoch besitzen die Modelle beide Eigenschaften, also die Erfüllung des Prinzips der maximalen Entropie und die Abhängigkeit des Modells von den Daten. Ferner sind beide Modelle in der Form von logarithmisch-linearen Modellen [69].

Als formeller Ausdruck lassen sich die Unterschiede zwischen generativen und deskriptiven Modellen folgendermaßen zeigen. Ein Markov-Modell kann die kombinierte Verteilung einer Beobachtung und des Zustandes des Systems wiedergeben [65], sofern beide Verteilungen bekannt sind. Das lässt sich ausdrücken durch

$$\begin{aligned} p(\mathbf{X}, \mathbf{S}) &= p(\mathbf{X}|\mathbf{S}) \cdot p(\mathbf{S}) \\ &= p(\mathbf{S}|\mathbf{X}) \cdot p(\mathbf{X}), \end{aligned}$$

wobei $p(\mathbf{X})$ die Verteilung der Beobachtungen und $p(\mathbf{S})$ die Verteilung der Zustandssequenz beschreibt. Diese Gleichheit ergibt sich aus den Regeln für die bedingte Wahrscheinlichkeit.

Im Falle der generativen Modelle wird sowohl $p(\mathbf{S})$ als auch $p(\mathbf{X}|\mathbf{S})$ als bekannt angenommen beziehungsweise wird aus den Trainingsdaten geschätzt. Trainiert man ein solches Modell, bestimmt man die Verbundwahrscheinlichkeit $p(\mathbf{X}, \mathbf{S})$ [65]. Ferner gilt

$$p(\mathbf{X}) = \int_{\mathbf{S}} p(\mathbf{X}|\mathbf{S}) dp(\mathbf{S}). \tag{4.1}$$

Das bedeutet, die Verteilung der Merkmale bestimmt sich als Randverteilung über die bedingte Wahrscheinlichkeit. Es lässt sich folglich die Verteilung der Beobachtungen aus den bekannten Größen her ableiten.

Im Falle der deskriptiven Modelle wird ausschließlich $p(\mathbf{S}|\mathbf{X})$ bestimmt. Alle anderen Verteilungen werden als unbekannt angenommen. Dadurch kann die Verbundwahrscheinlichkeit nicht bestimmt werden: Da insbesondere die Verteilung der Merkmale bei dem gegebenen Zustand $p(\mathbf{X}|\mathbf{S})$ nicht bekannt ist, ist die für generative Modelle wichtige Darstellung (4.1) nicht gegeben.

Um die Unterschiede zwischen HMMs, MEMMs und CRFs zu erläutern, wird zunächst die Markov-Kette $\mathbf{S}$ ohne die Messwerte betrachtet. In beiden Fällen, das heißt in generativen und deskriptiven Markov-Modellen, wird die Markov-Eigenschaft [28] in der Modellierung von $p(\mathbf{S})$ angenommen. Das bedeutet, dass die Zustände des Systems untereinander statistisch abhängig sind, im Falle von Zeitreihenanalysen, zu denen auch die Ereignisdetektion gehört, in einer zeitlichen Anordnung.

Bei der Aufgabe, die Verteilung eines einzelnen Zustandes innerhalb der Markov-Kette zu bestimmen, zeigen sich bereits deutlich die Unterschiede zwischen HMMs und MEMMs. Sehr oft wird in HMMs die Markov-Eigenschaft erster Ordnung [65] angenommen, bei der ein Zustand von dem vorherigen abhängig ist. Hieraus folgt unmittelbar, dass in dem Fall, dass die Wahrscheinlichkeiten für den vorhergehenden Zustand sowie sämtliche Übergangswahrscheinlichkeiten bekannt sind, sich aus Integration über diese vorherigen Zustände die Verteilung für den aktuellen Zustand berechnen lässt, das heißt

$$p(s(n)) = \int_{s(n-1)} p(s(n)|s(n-1))dp(s(n-1)),$$

da dieser aktuelle Zustand statistisch nur von den vorherigen abhängig ist.

Markov-Modelle höherer Ordnung werden seltener verwendet. Ein Grund ist, dass sie rechnerisch oft aufwendiger sind und somit weniger effizient: Ist ein Zustand von mehreren vorherigen Zuständen abhängig, so wird die Verteilung eines neuen Zustandes über die Integration der Verteilungen dieser vorherigen Zustände bestimmt [28], woran man den höheren Aufwand eines solchen Modells ersehen kann. Zugleich ist ersichtlich, dass das Markov-Modell im Fall des HMMs vollkommen autark entwickelt. Die Zustände sind ausschließlich von vorherigen abhängig, die Verteilung zu einem Zeitpunkt lässt sich ohne eine Beobachtung alleine aus der Information der Übergänge schätzen. Die Messwerte werden demnach für eine Evidienz für einen Zustand verwendet: In einem

HMM wird als generatives Modell angenommen, dass die Zustände unterschiedliche Wahrscheinlichkeiten haben, einen beobachteten Messwert zu generieren [65]. Anhand dieser Information verändert sich die Verteilung, in welchem Zustand sich das System zu dem Zeitpunkt der Messung befand. Dennoch war der Zustand zu dem Zeitpunkt, an dem die Messung stattfand, in diesem Modell fest.

In einem HMM wird angenommen, dass die Verteilung der Beobachtungen abhängig von den Zuständen ist [65]. Die Verteilung einer Beobachtung lässt sich über die Integration der Verteilung der Zustände bestimmen, also

$$p(\mathbf{x}(n)) = \int_{s(n)} p(\mathbf{x}(n)|s(n))dp(s(n)).$$

Die Anwesenheit der Beobachtung ist demnach für das eigentliche Modell nicht notwendig, auch wenn es sich um den messbaren Teil des Modells handelt: die Entwicklung des Markov-Modells lässt sich komplett autark beschreiben.

Bei den deskriptiven Modellen hingegen ist die Anwesenheit der Beobachtungen essentiell. Wird wieder die Verteilung eines einzelnen Zustandes an beliebiger Position der Markov-Kette gesucht, so ist dieses nur bei gegebenen Merkmalen möglich [41] mit

$$p(s(n)|\mathbf{X}) = \int_{s(n-1)} p(s(n)|s(n-1), \mathbf{X})dp(s(n-1)|\mathbf{X}). \tag{4.2}$$

Die Verteilung der Merkmale ist nicht notwendig für die Modellierung, da ausschließlich die gegebenen Merkmale in das Modell einfließen. Es wird also bestimmt, wie wahrscheinlich eine bestimmte Farbe bei einem Messwert ist, die Messwerte “steuern” demnach das Markov-Modell: Ein Messwert existiert vor dem Markov-Modell, der entsprechende Zustand nimmt erst durch die Anwesenheit des Messwertes eine Farbe an [41].

Diese Betrachtung zeigt, dass die Reihenfolge, in der in den Modellen die Knoten bestimmt werden, sich unterscheidet. In einem HMM wird zuerst der Zustand bestimmt und anschließend der Messwert, in einem MEMM wird zuerst der Messwert bestimmt, und nur mit ihm lässt sich die Verteilung des Zustandes bestimmen. Dieses zeigt, wie unterschiedlich die Modelle zu behandeln sind: HMMs bieten mehr Freiheiten bei der Anwendung, bedingen dafür jedoch mehr Annahmen, die in praktischen Problemen nicht immer gerechtfertigt sind. MEMMs sind in ihrer Anwendung etwas beschränkter, benötigen jedoch sichtlich weniger Annahmen, insbesondere über die Verteilung der Merkmale.

4.2 Das Prinzip der maximalen Entropie

Die Bestimmung einer Farbe für einen Zustand eines CRFs oder MEMMs basiert auf dem Prinzip der maximalen Entropie. Dieses Prinzip wird bereits seit langem bei Problemstellungen der Signalverarbeitung angewendet [23, 35]. Es hilft bei der Auswahl von statistischen Verteilungen für Probleme, bei denen nicht alle Informationen über das Signal vorliegen. Es lässt sich folgendermaßen zusammenfassen. Sofern alle präzise formulierten Informationen, das sind statistisch testbare Informationen, in einer Verteilungsfunktion verwendet wurden, ist nach dem Prinzip der maximalen Entropie diejenige Verteilungsfunktion zu wählen, die die höchste informationstheoretische Entropie aufweist [41, 54].

Eine Motivation für dieses Prinzip ist der Mangel an Informationen selbst. Die Entropie ist ein Maß für den Informationsgehalt. Berücksichtigt man alle Informationen, die einem gegeben sind, in einem Modell, können trotzdem noch mehrere Modelle zur Auswahl stehen. Dasjenige, das unter diesen Modellen, die noch wählbar sind, die höchste informationstheoretische Entropie enthält, ist demnach das Optimum hinsichtlich der Vermeidung nicht in der Datenmenge gegebener Informationen. Dieses Prinzip ist eine Empfehlung und somit nicht beweisbar. Dennoch liefert es Anhaltspunkte über die Art, nach der mit Informationen und Annahmen umgegangen werden kann. Eine Analyse der Schätzung der dazugehörigen Parameter befindet sich in [45], insbesondere wird dort der Zusammenhang zur Kullback-Leibler-Divergenz [36] gezeigt. Angelehnt hieran wird im folgenden eine Herleitung der Verteilung gezeigt, die zugleich die Grundlagen der nachfolgenden Verfahren bietet.

Gegeben seien eine der Einfachheit halber diskrete Menge von Beobachtungen $\mathbf{X}$ und entsprechende Merkmale $\phi_i(\mathbf{x}(n))$. Diese Beobachtungen sind verteilt nach einer unbekannten Dichte q, das heißt

$$\mathbf{x}(n) \sim q, n = 1, 2, \ldots. \tag{4.3}$$

Der Erwartungswert der extrahierten Merkmale wird demnach bestimmt durch

$$E_q\{\phi_i\} = \sum_{\mathbf{x} \in \mathbf{X}} q(\mathbf{x})\phi_i(\mathbf{x}).$$

Dieser Erwartungswert ist exakt, sofern die Dichte q bekannt ist. Da sie als unbekannt

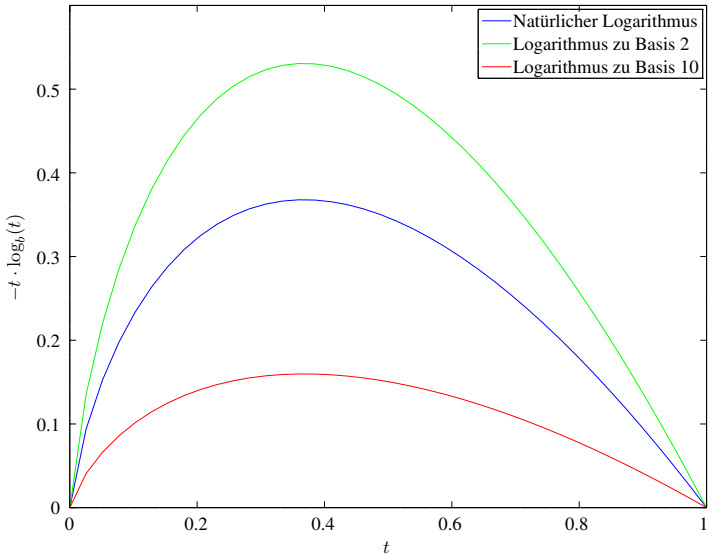


Abbildung 4.1: Graphische Darstellung für Entropie mit unterschiedlicher Basis. Der Wert der Entropie selbst ist abhängig von der Basis, das Maximum ist für jede Basis identisch. Für die hier vorgestellten Methoden wird der natürliche Logarithmus verwendet.

angenommen wird, wird eine (nicht notwendigerweise identische) Verteilung p gesucht, für die gilt

$$E_p\{\phi_i\} = E_q\{\phi_i\}, \tag{4.4}$$

für alle i, und deren informationstheoretische Entropie

$$H(p) = -\sum_{\mathbf{x}\in\mathbf{X}} p(\mathbf{x}) \log(p(\mathbf{x})) \tag{4.5}$$

maximal ist. Hierbei wurde der natürliche Logarithmus und somit die physikalische Entropie verwendet. Dieses hat historische Gründe, da dieser bei CRFs üblicherweise verwendet wird [41, 54]. Für die Wahl der Verteilung hat dieses keinen Einfluss, wie in Abbildung 4.1 zu sehen ist. Ferner gilt

$$\sum_{\mathbf{x}\in\mathbf{X}} p(\mathbf{x}) = 1, \tag{4.6}$$

da es sich bei p um eine Wahrscheinlichkeitsfunktion handelt.

Die aus (4.4), (4.5) und (4.6) gebildete Lagrange'sche Funktion [21] des Problems, die Entropie zu maximieren und die extrahierten Merkmale zu berücksichtigen, ist demnach

$$\Lambda(p, \lambda, \beta) = \beta(p(\mathbf{x}) - 1) - \sum_{\mathbf{x} \in \mathbf{X}} p(\mathbf{x}) \log(p(\mathbf{x})) + \sum_i \lambda_i (p(\mathbf{x})\phi_i(\mathbf{x}) - q(\mathbf{x})\phi_i(\mathbf{x})), \tag{4.7}$$

wobei λ_i und β Lagrange'sche Multiplikatoren sind. Diese verbinden die Nebenbedingungen mit der Zielfunktion. Abgeleitet nach $p(\mathbf{x})$, also der Änderung einer einzelnen Likelihood innerhalb der Gesamtverteilung, und für den Optimierungsschritt zu 0 gesetzt ist dieses

$$\frac{\delta\Lambda(p, \lambda, \beta)}{\delta p(\mathbf{x})} = \beta - \frac{p(\mathbf{x})}{p(\mathbf{x})} - \log(p(\mathbf{x})) + \sum_i \lambda_i \phi_i(\mathbf{x}) \stackrel{!}{=} 0. \tag{4.8}$$

Aufgelöst ergibt die Gleichung

$$\log(p(\mathbf{x})) = \beta - 1 + \sum_i \lambda_i \phi_i(\mathbf{x})$$

und somit

$$p(\mathbf{x}) = \exp(\beta - 1) \cdot \exp\left(\sum_i \lambda_i \phi_i(\mathbf{x})\right). \tag{4.9}$$

Der Lagrange'sche Multiplikator β lässt sich mit der Bedingung ersetzen, dass p eine Verteilung ist. Sei $Z_\lambda = \exp(1 - \beta)$. Dieser Wert ist konstant und für jede Wahrscheinlichkeit eines Messwertes gleich, somit gilt aufgrund von (4.6)

$$\begin{aligned} 1 &= \sum_{\mathbf{x} \in \mathbf{X}} \frac{1}{Z_\lambda} \exp\left(\sum_i \lambda_i \phi_i(\mathbf{x})\right) \\ &= \frac{1}{Z_\lambda} \sum_{\mathbf{x} \in \mathbf{X}} \exp\left(\sum_i \lambda_i \phi_i(\mathbf{x})\right) \end{aligned} \tag{4.10}$$

und demnach

$$Z_\lambda = \sum_{\mathbf{x} \in \mathbf{X}} \exp\left(\sum_i \lambda_i \phi_i(\mathbf{x})\right). \tag{4.11}$$

Offensichtlich lässt sich damit der Lagrange'sche Multiplikator β berechnen, das ist in

diesem Fall jedoch unnötig, da ausschließlich die Verteilung gesucht ist. Die Berechnung der anderen Lagrange'schen Multiplikatoren λ_i entspricht dem Bestimmen der konkreten Verteilung und somit dem Training des Modells. Es gilt somit

$$p(\mathbf{x}) = \frac{1}{Z_\lambda} \exp\left(\sum_i \lambda_i \phi_i(\mathbf{x})\right). \tag{4.12}$$

Dies wird im Folgenden *Verteilungsfunktion maximaler Entropie* genannt. Die Funktion $\exp\left(\sum_i \lambda_i \phi_i(\mathbf{x})\right)$ selbst ist eine nicht-normalisierte Verteilung:

Definition 4.1 (Nicht-normalisierte Verteilung) *Die Funktion $\tilde{p}$ ist eine nicht-normalisierte Verteilung, wenn es ein $z \in \mathbb{R}^+$ gibt, sodass fast überall gilt*

$$\tilde{p} = z \cdot p \tag{4.13}$$

und p eine Verteilung ist.

Die Verteilungsfunktion der maximalen Entropie bildet die Basis für die folgenden Modelle. Dabei wird die Verteilung eines Graphen, der statistisch abhängig von den Messwerten ist, bestimmt; das bedeutet, dass die Merkmalsextraktion eine Verbindung zwischen den Farben und den Messwerten herstellt. Dafür wird eine Funktion ϕ aufgestellt, die sowohl die Nachbarn im Zustandsgraphen als auch die gemessenen Merkmale berücksichtigt. Dabei wird im Folgenden zunächst eine sehr allgemeine Form dieser Funktion angenommen, erst später wird eine Konkretisierung dieser Merkmalsfunktion benötigt. Dadurch kann das CRF als allgemeines Modell besprochen werden, Erweiterungen und Spezialisierungen werden in der Darstellung voneinander getrennt.

4.3 Bedingte Zufallsfelder

CRFs sind als graphische Modelle vorgestellt worden [41]. Das bedeutet, dass sie sowohl aus einer graphischen als auch einer statistischen Komponente bestehen. Beide Aspekte sind für die CRFs von essentieller Bedeutung. Für eine Basis werden zunächst vor allem die graphischen Betrachtungen diskutiert.

Der graphische Aspekt beinhaltet zwei Darstellungen. Die erste Darstellung sind die Übergangsgraphen. Diese entsprechen dem Übergangsgraphen eines HMMs, wie sie bereits gezeigt wurden. Der zweite Graph stellt den Zusammenhang zwischen den Zuständen und Merkmalen her. Die Beziehungen der Knoten sind in Abbildung 4.2

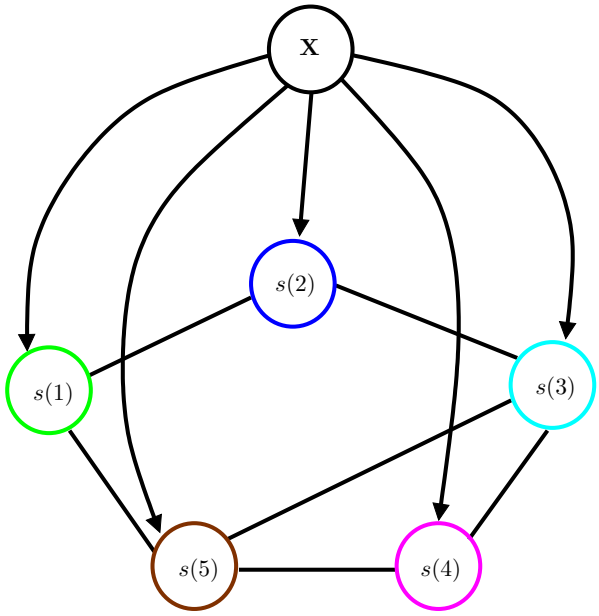


Abbildung 4.2: In einem allgemeinen CRF existiert zunächst keine Beschränkung bezüglich der Beziehungen zwischen den Zuständen. Die Messwerte (schwarzer Kreis) beeinflussen, welche Farbe die Zustände annehmen. Sie sind unabhängig vom Markov-Feld, können also auch mit unbekannten Prozessen generiert worden sein.

zu sehen. Ein wichtiger Bestandteil des CRFs ist dabei, dass die Beobachtungen die Verteilung der Zustände bedingen, anders als in einem HMM, bei dem die Zustände unterschiedlichen Verteilungen entsprechen, wobei angenommen wird, dass aus diesen die Messwerte gezogen wurden.

HMMs und lineare CRFs, also CRFs mit einer linearen Anordnung der Zustände, haben ebenfalls die zeitliche Ordnung gemein; diese ist in beiden Modellen diskret, das heißt, die Messungen werden zu diskreten Zeitpunkten angenommen. Für ein CRF ist ein Beispiel in Abbildung 4.3 zu sehen. Da es sich bei der Ereignisdetektion um eine Zeitreihenanalyse handelt, ist auch dieses Modell von zentraler Bedeutung, da somit die Zustände eine zeitliche Ordnung abbilden können.

Jeder Zustand nimmt eine Farbe an. Üblicherweise wird die Wahrscheinlichkeit für diese Farbe mittels des CRFs bestimmt, nicht die Farbe selbst; eine einzelne Farbe lässt sich zum Beispiel dadurch bestimmen, dass dem Zustand die Farbe mit der höchsten Wahrscheinlichkeit innerhalb der Sequenz zugeordnet wird. Dieses Vorgehen entspricht dem bekannten Viterbi-Algorithmus aus HMMs [65] und ist auch für CRFs und MEMMs üblich.

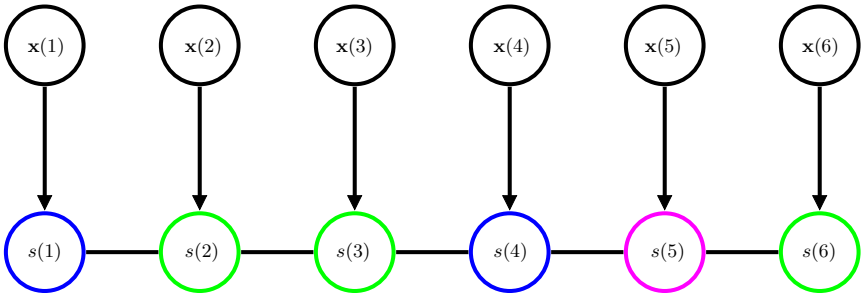


Abbildung 4.3: Darstellung eines linearen CRFs. Dieses Modell kann eine zeitliche Abfolge von Messungen modellieren. Ein Zustand ist abhängig von seinen beiden Nachbarn sowie dem Messwert. Durch die Abhängigkeitsstruktur wird das gesamte Zufallsfeld in einem Schritt bestimmt und ist daher endlich groß.

Der wesentliche Unterschied aus graphentheoretischer Sicht zwischen linearen CRFs, MEMMs und HMMs liegt, wie bereits kurz angesprochen, in der Anordnung der Graphen, die den zeitlichen Verlauf beschreiben. Ein Beispiel für ein lineares CRF befindet sich in Abbildung 4.3, für ein MEMM ist eine solche Darstellung in Abbildung 4.4 zu sehen. In diesen Graphen wird der Einfluss eines Knotens auf einen anderen dargestellt: Hat ein Knoten eine ausgehende oder ungerichtete Kante, so beeinflusst die Färbung dieses Knotens einen anderen; hat sie nur eingehende Kanten, wird er zwar von den Färbungen anderer Knoten beeinflusst, beeinflusst hingegen nicht selbst einen anderen Knoten. Statistisch gesehen entspricht diese Beziehung einer Abhängigkeit. Die Wahrscheinlichkeit, mit der ein Zustand eine Farbe annimmt, ist abhängig von den Verteilungen der Zustände entlang der eingehenden Kanten.

Diese statistische Betrachtung von CRFs, MEMMs und HMMs ist mit der graphischen Betrachtung verknüpft, die die Beziehung der Zustände beschreibt [41, 54, 65]. Im Folgenden wird die statistische Verteilung genauer betrachtet.

4.3.1 Statistische Betrachtung

Neben der graphischen Darstellung, also der Verbindung aus Knoten sowie gerichteten und ungerichteten Kanten innerhalb des Graphen, besitzen CRFs und MEMMs ebenso eine statistische Komponente. Diese beiden Modelle bilden einen Zusammenhang zwischen mehreren Zufallsvariablen. Jeder Knoten des Graphen entspricht dabei einer der Zufallsvariablen. Jede der Zufallsvariablen besitzt eine eigene Verteilung und eigene Abhängigkeiten. Ein solches Geflecht von Zufallsvariablen nennt man *Bayes'sches*

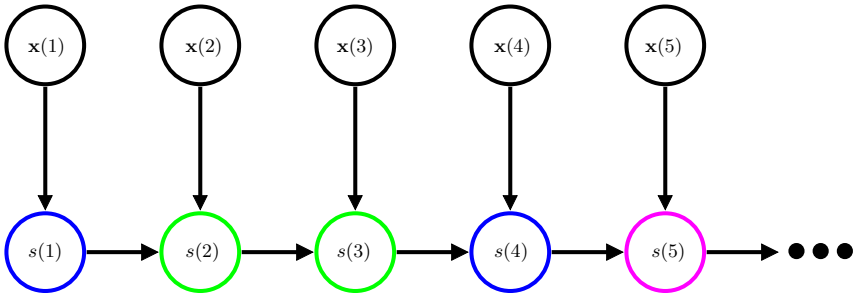


Abbildung 4.4: Bei MEMMs wird nur die lineare Form angenommen [53]. Ein neuer Zustand ist abhängig vom vorherigen sowie dem aktuellen Messwert. Durch die Abhängigkeitsstruktur ist es nicht notwendig, dass die Messreihe zeitlich begrenzt ist.

Netzwerk [3, 17]. CRFs sind somit ein Spezialfall eines Bayes'schen Netzes, ebenso wie andere Markov-Modelle.

Die Verteilungen der Zustände und der Messwerte werden unterschiedlich behandelt. Bei den Messwerten wird eine unbekannte Verteilung angenommen. Das bedeutet, dass ebenfalls eventuelle Abhängigkeiten zwischen den Merkmalen nicht bekannt sein müssen, um ein CRF beziehungsweise ein MEMM auszuwerten. Dadurch wird auch die oft getroffene Annahme über die Unabhängigkeit der Merkmale nicht benötigt. Dieses ist ein großer Vorteil dieser Modelle, der insbesondere bei der Ereignisdetektion wichtig sein kann.

Die Klasse der Zustände wird anders als bei HMMs nicht als elementarer Bestandteil der Entstehung der Messwerte betrachtet. Im Prinzip kann daher das CRF beziehungsweise MEMM als vollständig künstlich betrachtet werden: Natürliche Messwerte können direkt für das Modell verwendet werden, es dient ausschließlich der Analyse der Messwerte. Daher werden die Modelle derart gestaltet, dass sie auf Messwerten basieren. Statistisch gesehen entspricht das einer Abhängigkeit zu den gegebenen Messwerten.

Ein allgemeines CRF besitzt keine vorgegebene Struktur der Zustände, diese kann eigenständig bestimmt werden. Bei der Bestimmung der Verteilung eines Zustandes müssen die Nachbarschaftsbeziehungen berücksichtigt werden. Allgemein kann ein solches CRF demnach wie folgt auf Cliquen [77] definiert werden.

Definition 4.2 (Bedingte Zufallsfelder) *Sei* $\mathbf{S}$ *ein Zufallsfeld (der Graph der Zustände) und seien* $\mathbf{X}$ *gegebene Größen (Daten). Dann definiert* $p(\mathbf{S}|\mathbf{X})$ *ein* bedingtes Zufallsfeld *[28, 41], wenn gilt*

$$p(\mathbf{S}|\mathbf{X}) = \frac{1}{Z(\mathbf{X})} \prod_{C} \exp(Q(C, \mathbf{X})),$$

wobei C *eine Clique von* $\mathbf{S}$ *ist.* $Z(\mathbf{X})$ *ist eine Normierungskonstante, sodass* $p(S|\mathbf{X})$ *eine Wahrscheinlichkeit ist, das heißt, die Summe von* $p(\mathbf{S}|\mathbf{X})$ *über alle möglichen Instanzen des Zufallsfeldes* $\mathbf{S}$ *ist 1. Die Funktion* $\exp(Q(C, \mathbf{X}))$ *wird* Potentialfunktion *genannt [28]. Für die Potentialfunktion gilt*

$$Q(C, \mathbf{X}) = \lambda^{\top} \Phi(C, \mathbf{X}),$$

wobei $\lambda \in \mathbb{R}^d$ *ein Gewichtsvektor ist und* $\Phi(C, \mathbf{X}) \in \mathbb{R}^d$.

Die erste Bedingung in Definition 4.2 bedeutet, dass das CRF ein statistisch abhängiges Markov-Modell ist [28]. Eine ähnliche Cliquen-Faktorisierung existiert für jeden ungerichteten Markov-Graphen [77]. Die zweite Bedingung beschreibt die Eigenschaften einer Potentialfunktion. Diese lässt sich ausdrücken als eine logarithmisch-lineare Funktion [69], also eine Funktion, deren Logarithmus eine gewichtete Summe einzelner, reelwertiger und linearer Funktionen ist (die Komponenten der reell- und vektorwertigen Funktion Φ). Da der Logarithmus von $p(\mathbf{S}|\mathbf{X})$ ebenfalls eine Linearkombination dieser Funktionen ist, ist auch das CRF insgesamt eine logarithmisch-lineare Funktion.

Ein Zustand eines CRFs hängt von seinen Nachbarn ab, das sind alle Knoten, die adjazent zu ihm sind, also zu seiner Clique gehören, wie in Definition 4.2 erläutert. Ein Unterschied in der Behandlung der Zustände bezüglich der Daten ist, dass die Zustände durch ungerichtete Kanten verbunden sind, wohingegen sie nur eingehende Kanten haben, die mit den Daten verbunden sind. Das heißt, jeder Zustand ist abhängig von seinen Nachbarzuständen und den Daten, aber die Daten werden als unabhängig von den Zuständen angenommen. Das CRF wird somit von den Daten bestimmt; das Markov-Feld ist in Gänze abhängig von den Daten.

Eine unmittelbare Folge ist, dass die Daten nicht modelliert werden. Daraus folgt, dass eine Bestimmung der Verteilung der Daten unnötig ist und nur selten angewandt wird. Im Gegensatz zu CRFs ist dieses bei HMMs ein zentraler Bestandteil des Algorithmus und muss in jedem Fall stattfinden [65].

Ein wichtiger Spezialfall ist somit ein lineares CRF, bei dem die Zustände in einer

Markov-Kette angeordnet sind. Nicht alle Markov-Felder können direkt aus den Messwerten bestimmt werden, das heißt, deren Verteilung ist nicht immer eindeutig, allerdings gehören lineare CRFs zu den bestimmbaren [28, 41].

Definition 4.3 (Bedingte Markov-Zufallsketten) *Sei* $\mathbf{S}$ *ein Zufallsfeld (der Graph der Zustände) und seien* $\mathbf{X}$ *gegebene Größen (Daten). Haben alle Zustände bis auf zwei jeweils zwei Nachbarn, die beiden separaten jeweils einen, so bilden sie eine Markov-Kette* $\mathbf{S} = s(1), s(2), s(3), \ldots, s(N)$*. Jeder der Zustände kann eine von* K *Farben annehmen, das heißt* $s(n) \in \{\zeta_1, \zeta_2, \ldots, \zeta_K\}$*. Jede Farbe kann mehrmals innerhalb der Sequenz vorkommen. Das CRF, das von* $\mathbf{X}$ *und* $\mathbf{S}$ *gebildet wird, wird* lineares CRF *genannt. Insbesondere gilt*

$$\begin{aligned} p(\mathbf{S}|\mathbf{X}) &= \frac{1}{Z(\mathbf{X})} \prod_{C} \exp(Q(C, \mathbf{X})) \\ &= \frac{1}{Z(\mathbf{X})} \prod_{n=1}^{N} \exp(\lambda^\top \Phi(s(n), s(n+1), \mathbf{X}, n)), \end{aligned} \tag{4.14}$$

das heißt, jede Clique wird anhand des Übergangs zwischen zwei Zuständen vollständig beschrieben, die Potentialfunktion durch den Übergang zwischen zwei Zuständen und die Position innerhalb der Markov-Kette.

In der Sequenz $\mathbf{S}$ in Definition 4.3 ist ein Zustand nicht enthalten, nämlich $s(N+1)$. Dieser Zustand wird zur geschlossenen Form des CRFs genommen. Er ist somit nur symbolisch zu sehen. Durch diese Formulierung sieht man, dass für ein lineares CRF die Übergänge zwischen den Zuständen eine zentrale Bedeutung besitzen [41, 65], die Zustände selbst sind als Folgerung zu betrachten. Die Auflösung in die Übergänge erfolgt dadurch, dass die maximale Cliquenzahl, das ist die Anzahl der Knoten in der größten Clique, gleich zwei ist.

Ein Beispiel, bei dem lineare CRFs nützlich sind, ist die Textanalyse [55]. Texte bestehen aus natürlicher Sprache, somit sind Worte abhängig voneinander. Da die Abhängigkeiten sehr weitreichend sein können, zum Beispiel in längeren Sätzen, sind sie nur schwer direkt zu erfassen und als Markov-Modell zu modellieren. Insbesondere erhöht sich die Anzahl der notwendigen Berechnungen mit der Ordnung des Markov-Modells [28].

Die Anordnung des CRFs lässt sich also für dieses Problem unterschiedlich motivieren. Einerseits ist das Modell eine der einfachen Formen des CRFs, deren Bestimmung möglich ist. Andererseits bildet eine lineare Kette auch einen linearen Zusammenhang

ab, wie sie bei zeitlich oder allgemeiner linear geordneten Daten natürlich sind. Hierbei wird jedem Abschnitt (zum Beispiel bei zeitdiskreten Signalen jedem Datenpunkt $\mathbf{x}(n)$ oder jedem Wort im Text) ein Zustand $s(n)$ zugeordnet.

Das CRF lässt sich somit auf unterschiedlichen Ebenen betrachten. Auf einer globalen Ebene betrachtet man das gesamte Markov-Modell an sich, sowie die Abhängigkeiten zu gegebenen Größen. Auf der Ebene der Cliquen betrachtet man separat die Abhängigkeiten eines jeden Zustands zu den Nachbarn. Die Potentialfunktion ist mittels einer gewichteten Summe elementarer Eigenschaften, also der Merkmalsextraktion und den Kanten zu den Nachbarn, beschrieben.

Auf jeder dieser Ebenen lässt sich ein CRF für die jeweiligen Aufgaben anpassen. Auf der globalen Ebene konstruiert man ein Modell, das abhängig von den Daten ist und sie dadurch interpretiert. Zum Beispiel kann man ein lineares CRF für sequenzielle Daten konstruieren. Auf der Ebene der Clique kann man zum Beispiel Regeln für die Übergänge festlegen. In praktischen Anwendungen lassen sich diese Regeln zumeist aus den Daten schätzen. Unter anderem wird dieses im Training festgelegt. Auf der Ebene der Potentialfunktion entwirft man insbesondere die Merkmalsextraktion, also die vektorwertige Funktion Φ. Da diese sehr wichtig für das Gesamtkonzept ist, wird im Folgenden auf sie detailliert eingegangen. Vor allem ist sie abhängig von den Übergängen, mit denen man das CRF entwirft. Das heißt, hat man den Graphen für das CRF entworfen, entwirft man im nächsten Schritt die Funktion Φ. Mit dieser Funktion trainiert man das CRF. Das heißt, der Gewichtsvektor λ, der zur Potentialfunktion gehört, wird aus gegebenen Daten geschätzt. Mit diesem geschätzten Gewichtsvektor ist ein Markov-Modell zu der Datensequenz $\mathbf{X}$ gegeben. Da es sich bei der Ereignisdetektion um eine Zeitreihenanalyse handelt, werden als Design zumeist lineare CRFs verwendet; ausschließlich die Merkmalsextraktion ist somit noch notwendig, um das CRF komplett zu beschreiben.

Ist für eine Merkmalssequenz die Zustandssequenz nicht bekannt, so kann sie mittels eines gegebenen CRF geschätzt werden. Hierfür ist insbesondere die Sequenz maximaler Wahrscheinlichkeit interessant, also

$$\hat{\mathbf{S}} = \arg\max_{\mathbf{S}} p(\mathbf{S}|\mathbf{X};\lambda). \tag{4.15}$$

Die Bestimmung dieser Sequenz kann mittels eines Viterbi-Algorithmus erfolgen, ist jedoch unter Umständen aufwendig [65].

Zur Ereignisdetektion ist oft nicht die Verteilung der Gesamtsequenz interessant, sondern die Wahrscheinlichkeit jedes einzelnen Zustandes, diese ist

$$\hat{s}(n) = \arg\max_{\zeta} P(s(n) = \zeta | \mathbf{X}; \lambda). \tag{4.16}$$

Die Sequenz der Zustände mit der höchsten individuellen Wahrscheinlichkeit ist nicht unbedingt identisch mit der Sequenz der höchsten Wahrscheinlichkeit, wie sie in (4.15) gefordert ist. Insbesondere können so Sequenzen generiert werden, die nach einer Betrachtung des Übergangsgraphen nicht möglich sind [5, 65]. Ist dieser jedoch vollständig, so ist diese Sequenz eine Näherung an (4.15).

Zu einer anderen Sequenz $\mathbf{X}'$ ist ebenfalls die Verteilung $p(\mathbf{S}'|\mathbf{X}'; \lambda)$ gegeben; zu jeder Datensequenz kann eine entsprechende Zustandssequenz maximaler Wahrscheinlichkeit generiert oder bei Merkmalssequenzen gleicher Länge gehören die Zustandssequenzen verglichen werden. Ebenso ist es erlaubt, die Verteilung jedes einzelnen Zustandes $s(n)$ zu bestimmen. Dieses ermöglicht es, neue Datensequenzen mit einem CRF zu analysieren, was die Grundlage zur Ereignisdetektion liefert. Das Problem der Ereignisdetektion mittels eines CRF ist somit, Ereignisse in einer Zustandssequenz zu detektieren, was einfacher sein kann als in den Sequenzen von Messwerten $\mathbf{X}$, insbesondere, wenn die Verteilung von $\mathbf{x}(n)$ nicht bekannt ist. Das CRF kann demnach als Abstraktion von den realen Messwerten betrachtet werden. Führt man sich das Beispiel eines heterogenen Sensornetzwerkes vor Augen, zeigt sich der große Vorteil dieses Vorgehens: anstelle sehr komplexer Verteilungen und großer Unsicherheiten bei Modellannahmen wird eine einfacher zu interpretierende Zustandssequenz analysiert.

Vor dem CRF wurden die MEMMs vorgestellt, bei denen die Übergänge nur in eine Richtung statistisch abhängig sind [54]. Dieses Modell zeichnet sich durch gerichtete Verknüpfungen der Zustände ab. CRFs beinhalten ungerichtete Graphen [41]. Um die Vorteile beider Methoden zu vereinen, wurde im Zusammenhang mit der Ereignisdetektion im Rahmen dieser Arbeit ein Hybrid-Graph [22] entworfen. Im Folgenden werden die drei Modelle separat betrachtet, um die Vor- und Nachteile darzulegen.

4.4 Ein-Schritt-Auswertung (Maximum-Entropy-Markov-Modelle)

Die schrittweise Auswertung von Markov-Modellen ist spätestens seit der Einführung der HMMs etabliert [65]. Dabei bilden die Zustände eine Kette, im Fall der Ereig-

nisdetektion in linearer Abfolge. Ist diese in zeitlicher Abfolge, so kann die Kette die Kausalität darstellen: in diesem Fall ist ein aktueller Zustand nur von vorherigen Zuständen abhängig, nicht von zukünftigen.

In HMMs werden grundsätzlich die Zustände als statistisch angesehen. Das heißt, es wird eine Wahrscheinlichkeit für jede Färbung des Zustandes bestimmt, nicht der konkrete Zustand selbst. Ist diese Instanz gewünscht, so wird nach einer Auswertung, zum Beispiel dem Viterbi-Algorithmus [65], eine Zustandssequenz aus diesen Wahrscheinlichkeiten bestimmt. Üblich ist auch, dass zur Bestimmung eines Zustandes die Wahrscheinlichkeiten der Parentalknoten verwendet werden, also die Knoten, von denen eine gerichtete Kante zum aktuellen Knoten ausgeht. Alternativ kann auch ausschließlich die Färbung mit der höchsten Wahrscheinlichkeit genommen werden.

Häufig wird angenommen, dass der aktuelle Zustand nur vom vorherigen abhängig ist, nicht von länger zurück liegenden. Dadurch wird die Information über in der Sequenz weiter entfernt liegende Zustände nur über die Wahrscheinlichkeit des vorherigen Zustandes weitergegeben. Diese Annahme wird häufig getroffen, da komplexere Modelle auch mit aufwendigeren Berechnungen verbunden sind [65].

MEMMs sind ebenfalls Markov-Modelle erster Ordnung mit gerichteten Kanten, wie HMMs [53]. Kein Zustand ist abhängig von zukünftigen oder von sehr weit zurückliegenden Zuständen. Hinzu kommt bei MEMMs die Abhängigkeit von den Messwerten. Dieses unterscheidet MEMMs von den HMMs in sofern, als dass bei der Modellvorstellung von HMMs angenommen wird, dass die Merkmale von den Zuständen abhängig sind. Das Modell der MEMMs wurde im Jahr 2000 vorgestellt [54] und kann als das Vorgängermodell zu CRFs betrachtet werden.

Die Wahrscheinlichkeit, dass zum Zeitpunkt n die Farbe ζ_k beobachtet wird, ist in einem MEMM gegeben durch

$$\begin{aligned} P(s(n) = \zeta_k | \mathbf{x}(1), \mathbf{x}(2), \ldots, \mathbf{x}(n)) = P(s(n) = \zeta_k | \mathbf{x}(n)) \\ \times P(s(n) = \zeta_k | \mathbf{x}(1), \mathbf{x}(2), \ldots, \mathbf{x}(n-1)), \end{aligned} \tag{4.17}$$

wobei gilt, dass die Vorhersage ausschließlich unter Verwendung der vergangenen Messwerte $\mathbf{x}(1), \mathbf{x}(2), \ldots, \mathbf{x}(n-1)$ bestimmt wird durch

$$\begin{aligned} P(s(n) = \zeta_k | \mathbf{x}(1), \mathbf{x}(2), \ldots, \mathbf{x}(n-1)) = \sum_{j=1}^{K} P(s(n) = \zeta_k | s(n-1) = \zeta_j) \\ \times P(s(n-1) = \zeta_j | \mathbf{x}(1), \mathbf{x}(2), \ldots, \mathbf{x}(n-1)). \end{aligned} \tag{4.18}$$

Es lässt sich dadurch die Wahrscheinlichkeit einer Farbe zu jedem Zeitpunkt rekursiv

bestimmen, da sich $P(s(n-1) = \zeta_j | \mathbf{x}(1), \mathbf{x}(2), \ldots, \mathbf{x}(n-1))$ in (4.18) äquivalent zu (4.17) bestimmen lässt. Diese Eigenschaft hat den Vorteil, dass damit die Sequenzierung, also die Auswertung einer Zeitreihe, auch echtzeitfähig ist, da zu der Auswertung der Verteilung der Zustände zum Zeitpunkt n nur die vorherigen Messungen benötigt werden, deren Information ferner durch die Zustandssequenz übertragen wird.

Ein weiterer Vorteil dieses Modells gegenüber HMMs wird an dieser Stelle deutlich. Bei der Bestimmung eines HMMs muss die Initialwahrscheinlichkeit, das ist die Verteilung des Zustandes $s(1)$, separat betrachtet werden [65]. Zu einer Schätzung dieser Initialwahrscheinlichkeit werden mehrere Sequenzen benötigt. Bei MEMMs wird diese Verteilung direkt modelliert durch $p(s(1)|\mathbf{x}(1))$, der Anteil der Übergangswahrscheinlichkeiten wird nicht betrachtet, da zu diesem Zeitpunkt noch kein Übergang stattgefunden hat. Dadurch bestimmen dieselben Parameter, die auch im späteren Verlauf der Sequenzierung verwendet werden, diese Initialwahrscheinlichkeit. Demnach wird zu ihrer Bestimmung auch nur eine Sequenz benötigt; der erste Zustand wird wie jeder folgende abhängig der Beobachtung bestimmt, nicht anhand einer abstrakten Initialverteilung.

MEMMs haben große Ähnlichkeit mit den in Definition 4.3 vorgestellten linearen CRFs. In MEMMs wird die Wahrscheinlichkeit, dass das Modell zu einem Zeitpunkt n eine spezielle Färbung ζ_k annimmt, definiert durch

$$P(s(n) = \zeta_k | \mathbf{x}(n), s(n-1)) = \frac{1}{Z(\mathbf{x}(n))} \exp(\lambda^\top \Phi(s(n-1), \zeta_k, \mathbf{x}(n))), \qquad (4.19)$$

wobei Φ eine vergleichbare Potentialfunktion wie in 4.3 ist und

$$Z(\mathbf{x}(n)) = \sum_{j=1}^{K} \exp(\lambda^\top \Phi(s(n-1), \zeta_j, \mathbf{x}(n))).$$

Wie bei linearen CRFs besteht somit das Entwerfen eines MEMMs dadurch, die Potentialfunktion zu bestimmen; die Gewichte λ werden trainiert und sind somit von konkreten Daten abhängig. Dieses entspricht der Verteilung der maximalen Entropie über die Zustände, wobei die Funktion, die die testbaren Informationen enthält, ebenfalls die Messwerte berücksichtigt, wie in Abschnitt 4.2 besprochen wurde.

In (4.19) ist der vorherige Zustand $s(n-1)$ als bekannt angenommen worden. In der Praxis ist dieser Zustand rekursiv bestimmt. Zur Auswertung eines MEMMs kann für

die Bestimmung der Wahrscheinlichkeit des aktuellen Zustandes der weiche Viterbi-Dekoder [54, 65] $P(s(n) = \zeta_k|\mathbf{x}(n)) = \alpha(n,k)$ genutzt werden mit

$$\tilde{\alpha}(n,k) = \begin{cases} \exp\left(\lambda^\top \Phi(s_0, \zeta_k, \mathbf{x}(1))\right), & \text{falls } n = 1 \\ \sum_{j=1}^{K} \alpha(n-1,j) \cdot \exp\left(\lambda^\top \Phi(\zeta_j, \zeta_k, \mathbf{x}(n))\right), & \text{sonst} \end{cases}$$

$$\alpha(n,k) = \frac{\tilde{\alpha}(n,k)}{\sum_{j=1}^{K} \tilde{\alpha}(n,j)}. \tag{4.20}$$

Diese Auswertung berücksichtigt auch, dass nicht jeder Zustand mit hoher Genauigkeit bestimmt werden kann. Ist ein einzelner Pfad, das heißt eine bestimmte Sequenz an Zuständen, erwünscht, kann dieser mittels das Viterbi-Algorithmus [65] bestimmt werden.

Es lässt sich zeigen [63], dass die Formulierung der MEMMs dem Prinzip der maximalen Entropie genügt. Das bedeutet, dass diese Verteilung die Information der Datenextraktion optimal nutzt, um das daran anschließende Markov-Modell zu bestimmen. Diese generelle Eigenschaft ist für die Ereignisdetektion sehr nützlich, da hierdurch keine unnötigen Annahmen über die Daten an sich gestellt werden müssen.

Die Struktur eines MEMM hat jedoch die Eigenschaft, dass zukünftige Messungen nicht betrachtet werden, selbst wenn diese schon vorliegen. Da solche Modelle vor allem angewendet werden, wenn die Messung $\mathbf{x}(n)$ alleine nicht genügt, um den Zustand $s(n)$ eindeutig zu bestimmen, kann dieses ein Nachteil sein; das heißt, es ließe sich der Zustand eventuell genauer bestimmen, wenn zukünftige Messungen berücksichtigt würden. Im Vergleich zu HMMs muss hierfür allerdings das Modell geändert werden. Während in HMMs diese Einbeziehung der zukünftigen Messungen durch die Bayes-Regel möglich ist [65], da in diesen Modellen die unbekannten Zustände die Messungen bestimmen und somit die Information aus den Messungen zurückgewonnen wird, haben in einem MEMM die zukünftigen Messungen auf vorherige Zustände keinen Einfluss. Werden folglich Informationen aus zukünftigen Messungen berücksichtigt, werden diese auch antikausal betrachtet. Die Kanten, die die Zustände verbinden, sind demnach ungerichtet. Dieses ist der Unterschied zwischen MEMMs und linearen CRFs.

4.5 Auswertung von bedingten Zufallsfeldern

Wie in Abschnitt 4.4 bereits erwähnt existiert ein starker Zusammenhang zwischen CRFs und MEMMs. Beides sind datengetriebene graphische Modelle, die auf dem Prinzip

der maximalen Entropie, siehe Abschnitt 4.2, beruhen [41, 54, 63]. Dennoch gibt es signifikante Unterschiede zwischen den Modellen.

Der wichtigste Unterschied ist, dass die Zustände in MEMMs durch gerichtete Graphen beschrieben werden. In diesen Modellen ist jeder Zustand nur von vorherigen statistisch abhängig. In CRFs ist das nicht der Fall: die Zustände in CRFs bilden ungerichtete Graphen. Das bedeutet, dass die Verteilung der Färbung zweier benachbarter Zustände gemeinsam beschrieben wird, wie in Definition 4.2 erläutert wird.

MEMMs hatten insbesondere eine rein lineare Struktur. Der für die Ereignisdetektion wichtigste Spezialfall hinsichtlich CRFs ist ebenfalls einer mit einer linearen Struktur, das sind die linearen CRFs, siehe Definition 4.3. Grundsätzlich sind CRFs nicht auf diese Form beschränkt.

Ein weiterer wichtiger Unterschied ist, dass die Zustände im Prinzip nicht ausschließlich von einem aktuellen Messwert abhängig sind, wie es in MEMMs der Fall ist. Stattdessen kann jeder Zustand von beliebig vielen Beobachtungen abhängig sein. Dieser Fall tritt jedoch in der Ereignisdetektion nicht oft ein, da sich diese Problemstellung oft damit beschäftigt, zu entscheiden, ob ein Messwert dem Normalfall oder dem Ereignis angehört. Dennoch sind auch hier Interpretationen über längere Zeiträume in einigen Fällen in die Betrachtung einzubeziehen.

Eine Diskussion über allgemeine CRFs befindet sich in [28]. Nicht alle CRFs besitzen bei gegebenen Merkmalen eine eindeutige Färbung; das heißt, nicht jede Anordnung eines beliebigen Markov-Feldes lässt sich bei gegebenen Merkmalen eindeutig färben. Der wichtige Spezialfall, die linearen CRFs, gehört allerdings nicht dazu, dieser lässt sich eindeutig bestimmen. Als Vereinfachung wird im Folgenden die Einschränkung angenommen, dass weiterhin zu jedem Messwert $\mathbf{x}(n)$ genau ein Zustand $s(n)$ existiert, entsprechend der Modellierung als MEMM. Dadurch werden komplexe Notationen vermieden. Es sei an dieser Stelle ausschließlich darauf verwiesen, dass der Vektor $\mathbf{x}(n)$ ebenfalls Messwerte enthalten kann, deren Messung bereits längere Zeit zurückliegt, und somit eine mögliche lange Verbindung zwischen den Merkmalen implizit auch mit dieser Einschränkung behandelt werden kann.

Sind in dem Graphen der Zustände die Knoten linear angeordnet, so haben alle Zustände zwei Nachbarn, bis auf den ersten und den letzten Zustand, $s(1)$ respektive $s(N)$. Sei $s(n)$, mit $1 < n < N$ ein Zustand dieser Sequenz. Dann sind wegen der Reihenfolge $p(s(n)|\mathbf{x}(1), \mathbf{x}(2), \ldots, \mathbf{x}(n))$ und $p(s(n)|\mathbf{x}(n+1), \mathbf{x}(n+2), \ldots, \mathbf{x}(N))$

statistisch unabhängig, da $s(n)$ den Abhängigkeitsgraphen teilt. Daraus ergibt sich, dass gilt

$$\tilde{p}(s(n)|\mathbf{X}) = \\ p(s(n)|\mathbf{x}(1), \mathbf{x}(2), \ldots, \mathbf{x}(n)) \cdot p(s(n)|\mathbf{x}(n+1), \mathbf{x}(n+2), \ldots, \mathbf{x}(N)), \tag{4.21}$$

wobei $\tilde{p}(s(n)|\mathbf{X})$ eine nichtnormalisierte Verteilung ist. Das bedeutet, die Verteilung des Knotens ist separierbar. Diese Eigenschaft wird genutzt, um die Färbung des Graphen zu bestimmen. Dabei werden die ungerichteten Kanten durch je zwei gerichtete dargestellt; dadurch entsteht ein vorwärts und ein rückwärts gerichteter Pfad.

Diese Pfade werden folgendermaßen definiert. Sei $\mathbf{X}$ die Sequenz der Messwerte, λ und $\Phi(s_1, s_2, \mathbf{X}, n)$ wie in (4.14). Dann ist der vorwärts gerichtete Pfad $\alpha(n, k)$ rekursiv definiert durch

$$\begin{aligned} \tilde{\alpha}(n,k) &= \begin{cases} \exp\left(\lambda^\top \Phi(s_0, \zeta_k, \mathbf{X}, 1)\right), & \text{falls } n = 1 \\ \sum_{j=1}^{K} \alpha(n-1, j) \cdot \exp\left(\lambda^\top \Phi(\zeta_j, \zeta_k, \mathbf{X}, n)\right), & \text{sonst} \end{cases} \\ \alpha(n,k) &= \frac{\tilde{\alpha}(n,k)}{\sum_{j=1}^{K} \tilde{\alpha}(n,j)}, \end{aligned} \tag{4.22}$$

beziehungsweise der rückwärts gerichtete Pfad $\beta(n, k)$ rekursiv definiert durch

$$\begin{aligned} \tilde{\beta}(n,k) &= \begin{cases} \exp\left(\lambda^\top \Phi(\zeta_k, s_0, \mathbf{X}, N+1)\right) & \text{falls } n = N \\ \sum_{j=1}^{K} \beta(n+1, j) \cdot \exp\left(\lambda^\top \Phi(\zeta_k, \zeta_j, \mathbf{X}, n+1)\right) & \text{sonst} \end{cases} \\ \beta(n,k) &= \frac{\tilde{\beta}(n,k)}{\sum_{j=1}^{K} \tilde{\beta}(n,k)}. \end{aligned} \tag{4.23}$$

$s_0 \notin \{\zeta_1, \zeta_2, \ldots, \zeta_K\}$ ist ein Symbol für eine Färbung, die kein Zustand innerhalb der Kette annehmen kann [28]. Es dient dazu, das Markov-Modell zu initiieren beziehungsweise zu terminieren und kann als Anfangs- respektive Endzustand betrachtet werden. Man beachte, dass der vorwärts gerichtete Pfad einem MEMM entspricht; der rückwärts gerichtete somit einem antikausalen MEMM.

Die Wahrscheinlichkeit, dass das lineare CRF zum Zeitpunkt n die Färbung ζ_k erhält, gegeben die Messwerte $\mathbf{X}$, ist somit [28, 41]

$$\tilde{P}(s(n) = \zeta_k|\mathbf{X}) = \alpha(n,k) \cdot \beta(n,k). \tag{4.24}$$

Somit lässt sich in einem linearen CRF sehr einfach und effizient die Färbung eines Zustandes bestimmen [28]. Allerdings ist in diesem Fall jede Messung der gesamten Zeitreihe zur Bestimmung des Zustandes verwendet, wie auch in Abbildung 4.5 graphisch dargestellt wird. Das hat den Nachteil, dass ein lineares CRF nicht direkt zur Ereignisdetektion verwendet werden kann, wenn die Messungen noch nicht vollständig sind, was zum Beispiel bei Überwachungsproblemen üblich ist. Bei diesen Problemen ist es für gewöhnlich erforderlich, dass die Auswertung parallel oder mit einer festen Verzögerung zur Messung stattfindet, da eine unmittelbare Reaktion erforderlich sein kann.

Der Vorteil hingegen ist, dass die Färbung eines jeden Zustandes genauer, das heißt mit der Betrachtung von mehr Informationen, bestimmt werden kann. Eine Messung kann durchaus zu unterschiedlichen Färbungen innerhalb der Sequenz führen; die präzise Färbung ist abhängig von Messungen, die mit späteren Zuständen korrelieren. Dieses ist ein wichtiger Vorteil von CRFs gegenüber MEMMs.

Solche Glättungsverfahren, wie durch (4.24) entstehen, sind auch bei HMMs üblich [65]. Bei MEMMs respektive CRFs führt dieser Schritt zu einer neuen Interpretation der Abhängigkeiten und dadurch zu einem neuen Modell.

Es ist wünschenswert, dass die erhöhte Präzision, mit der die Färbung eines Zustandes bestimmt werden soll, auch für die Ereignisdetektion genutzt wird. Durch die Betrachtung von mehr Informationen über die Nachbarschaftsbeziehungen der Zustände ist nach dem Prinzip der maximalen Entropie ein besseres Modell erreicht. Als Vorschlag, um diese Eigenschaft für eine Datenanalyse zu verwenden, die parallel zur Messung stattfindet, also ein echtzeitfähiges Verfahren, wurde im Rahmen dieser Arbeit ein entsprechender Hybrid-Graph entwickelt [51].

4.6 Sequenzielle Auswertung von Segmenten

Die beiden logarithmisch-linearen Modelle, die bisher besprochen wurden, die MEMMs und die CRFs, besitzen beide praktisch relevante Eigenschaften, die zur Ereignisdetektion genutzt werden sollen. Die Ein-Schritt-Auswertung in MEMMs wie in Gleichung

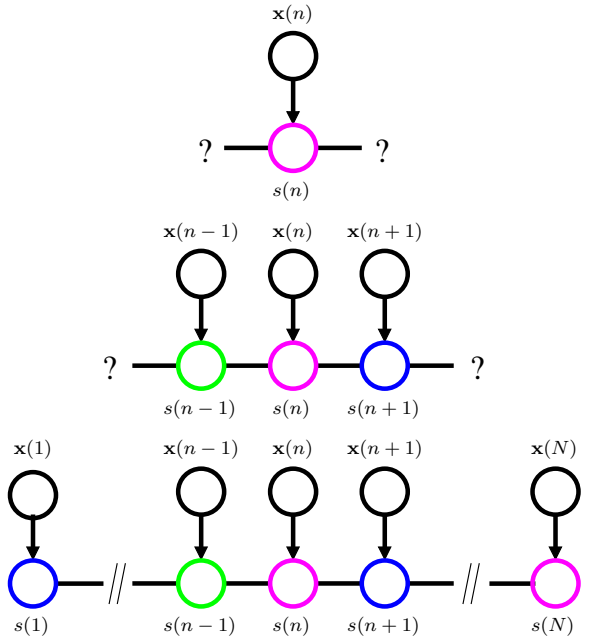


Abbildung 4.5: Zur Bestimmung der Farbe eines einzelnen Zustandes innerhalb des CRFs müssen die Färbungen seiner Nachbar-Zustände bekannt sein, ebenso die Messungen. Dieses setzt sich fort, sodass für die Bestimmung eines einzelnen Zustandes die gesamte Messsequenz bekannt sein muss.

(4.19) ist schnell und ermöglicht eine echtzeitfähige Auswertung, jedoch kann für die Bestimmung eine höhere Genauigkeit erwünscht sein. Bei CRFs werden Informationen der gesamten Messung verwendet. Das ermöglicht eine genauere Bestimmung der Zustände, dadurch müssen jedoch auch die letzten Messungen berücksichtigt werden. CRFs sind somit nicht direkt zur Echtzeitauswertung geeignet. Für die Ereignisdetektion ist ein solcher echtzeitfähiger Algorithmus sehr wichtig, da es Problemstellungen gibt, bei denen die Auswertung parallel zur Aufnahme durchgeführt wird. Hierfür wurde im Rahmen dieser Arbeit als Ergänzung ein Markov-Modell entwickelt, das beide Vorteile vereint [51]. Der dabei entwickelte Algorithmus wird im folgenden im Detail vorgestellt.

Dabei wird berücksichtigt, dass die Informationen, die einen Zustand beschreiben, auch im CRF oft nur sehr lokal bedeutend sind. Diese Informationen werden nur über die Verteilungen der Zustände weiter gegeben. Es ist daher anzunehmen, dass abhängig von den Informationen, die eine Messung bringt, nur kurze Messreihen für die ausreichende Bestimmung der Färbung eines Zustandes benötigt werden. Messungen, die lange vor oder nach einem betrachteten Zustand stattfinden, haben nur sehr bedingten Einfluss auf die Färbung eines Knotens, wohingegen Messungen kurz vor oder nach diesem Zustand sehr wichtig sein können. Insbesondere ist hier der kausale Zusammenhang zu

beachten: Sind die Messungen für die Entscheidung geeignet, werden nach Möglichkeit die notwendigen Informationen auch zeitnah im System sein. Das bedeutet, dass weder Messungen, die lange zurück liegen, auf ein Ereignis hindeuten, ohne dass dieses auch in aktuelleren Messungen zu identifizieren ist, als auch, dass die Messung nicht wesentlich verzögert durchgeführt wird. Damit ist der kausale Zusammenhang wichtiger als der antikausale, und Informationen aus früheren Zuständen müssen für die Bestimmung einer Färbung berücksichtigt werden, aus späteren Messungen reicht eine kürzere Sequenz, die den Verzug zur Bestimmung einer Zustandsfärbung führt. Im Folgenden wird diese Eigenschaft *lokal konzentrierte Information* genannt.

Das Modell des Zustandsgraphen, das als Erweiterung zu MEMMs und CRFs entwickelt wurde, um diesen Umstand zu berücksichtigen, ist ein Hybrid-Graph. Das heißt, in dem Graphen gibt es sowohl gerichtete als auch ungerichtete Kanten. Diese Kanten beschreiben, wie die Information der Messwerte in dem Graphen verteilt werden. Die ungerichteten Kanten beschreiben eine lokale Gruppe innerhalb des Graphen, während die gerichteten Kanten eine Weiterführung der Information in nur eine Richtung beschreiben.

Dafür werden zuerst Segmente der Datensequenz $\mathbf{X}$ betrachtet. Diese Segmente besitzen immer eine feste Länge T. Das erste Segment ist somit $\mathbf{X}^1 = \mathbf{x}(1), \mathbf{x}(2), \ldots, \mathbf{x}(T)$. Zu diesem Segment wird ein lineares CRF bestimmt, das heißt, die Zustandssequenz $\mathbf{S}^1 = s(1), s(2), \ldots, s(T)$, wie in (4.15) beschrieben. Das heißt, dass zunächst die Wahrscheinlichkeit für jede Farbe zu jedem Zeitpunkt dieses Segmentes, also $p(\mathbf{S}^1|\mathbf{X}^1)$ wie in (4.24), bestimmt wird. Anschließend wird die Sequenz der höchsten Likelihood wie in (4.24) bestimmt. Dadurch ist dieses Segment äquivalent zu einem CRF gefärbt, die lokalen Informationen wurden ausgenutzt.

Für die meisten Zustände ist unter der Annahme, dass die Information lokal konzentriert ist, diese Bestimmung ausreichend, das heißt, auch durch weitere Messung wird sich die Färbungen der Zustände mit hoher Wahrscheinlichkeit nicht ändern. Die einzigen Zustände, deren Färbung sich unter Berücksichtigung zukünftiger Messungen ändern können, sind diejenigen mit hohem Zeitindex, also $s(T - \tau + 1)$, $s(T - \tau + 2)$, $\ldots$, $s(T)$, wobei $0 < \tau < T$ gilt.

Sei nun $\mathbf{X}^2 = \mathbf{x}(T - \tau + 1), \mathbf{x}(T - \tau + 2), \ldots, \mathbf{x}(2 \cdot T - \tau)$ ein weiteres Segment der Daten. Dieses Segment hat mit dem ersten in der Gesamtsequenz genau die Messwerte gemein, für die die entsprechenden Zustände andere Farben annehmen können. Mit $\mathbf{x}^l(n) = \mathbf{x}(n + (l-1) \cdot T - \tau)$ für $l > 1$ ist $\mathbf{X}^2 = \mathbf{x}^2(1), \mathbf{x}^2(2), \ldots, \mathbf{x}^2(T)$, und allgemein sind $\mathbf{X}^l = \mathbf{x}^l(1), \mathbf{x}^l(2), \ldots, \mathbf{x}^l(T)$ Segmente, für die ein Zustandsgraph bestimmt

werden soll. Mit $s^l(n) = s(n + (l-1) \cdot T - \tau)$ sind die linearen Markov-Felder $\mathbf{S}^l$ und deren Likelihood $p(\mathbf{S}^l|\mathbf{X}^l)$. Es lassen sich demnach für alle Segmente sequenziell Zustandsgraphen bestimmen.

Hierbei ist zu beachten, dass für die Färbung der Knoten zu Beginn der Segmente mit Index $l > 1$, also genau die in mehreren Segmenten vorhandenen Zustände, die gesamte lokal konzentrierte Informationen verwendet wird. Ihre Bestimmung ist somit nicht genauer als ohne die Überlappung der Segmente. Wünschenswert ist, dass Informationen aus dem vorher ausgewerteten Segment in das nächste übernommen werden.

Diese Informationen werden dem vorwärts gerichteten Pfad zur Bestimmung der Färbung übergeben; der rückwärts gerichtete Pfad trägt Informationen aus später gemessenen Daten zu den Knoten, deren Färbung als nicht genau angenommen wird. Der rückwärts gerichtete Pfad, der für die von den Segmenten gebildeten CRFs genutzt wird, ist demnach

$$\begin{aligned}
\tilde{\beta}^l(n,k) &= \begin{cases} \exp\left(\lambda^\top \Phi(\zeta_k, s_0, \mathbf{X}^l, N+1)\right) & \text{falls } n = T \\ \sum_{j=1}^{K} \beta^l(n+1,j) \cdot \exp\left(\lambda^\top \Phi(\zeta_k, \zeta_j, \mathbf{X}^l, n+1)\right) & \text{sonst} \end{cases} \\
\beta^l(n,k) &= \frac{\tilde{\beta}^l(n,k)}{\sum_{j=1}^{K} \tilde{\beta}^l(n,k)},
\end{aligned} \tag{4.25}$$

wobei der obere Index derjenige des Segments ist. Dieser Pfad entspricht dem üblichen Pfad in einem CRF, das für das aktuelle Segment $\mathbf{X}^l$ bestimmt wird.

Der vorwärts gerichtete Pfad unterscheidet sich von einer normalen CRF. Diesem wird die Information aus vorherigen Segmenten übergeben. Dieses erfolgt durch die Übergabe der Information der Likelihood des letzten sicher bestimmten Zustands aus dem vorherigen Segment:

$$\begin{aligned}
\tilde{\alpha}^l(n,k) &= \begin{cases} p(s^{l-1}(T-\tau)|\mathbf{X}^1, \mathbf{X}^2, \ldots, \mathbf{X}^l) \cdot \exp\left(\lambda^\top \Phi(s_0, \zeta_k, \mathbf{X}^l, 1)\right), & \text{falls } n = 1 \\ \sum_{j=1}^{K} \alpha^l(n-1,j) \cdot \exp\left(\lambda^\top \Phi(\zeta_j, \zeta_k, \mathbf{X}^l, n)\right), & \text{sonst} \end{cases} \\
\alpha^l(n,k) &= \frac{\tilde{\alpha}^l(n,k)}{\sum_{j=1}^{K} \tilde{\alpha}^l(n,j)}.
\end{aligned} \tag{4.26}$$

Die (nicht-normalisierte) Wahrscheinlichkeit für einen Zustand des CRFs, das durch das Segment $\mathbf{X}^l$ gebildet wird, eine Färbung ζ_k anzunehmen, ist demnach gegeben mit

$$\tilde{P}(s^l(n) = \zeta_k|\mathbf{X}^1, \mathbf{X}^2, \ldots, \mathbf{X}^l) = \alpha^l(n,k) \cdot \beta^l(n,k). \tag{4.27}$$

Dadurch sind die Verteilungen für alle Zustände bestimmt. Die Berücksichtigung der historischen Daten ist hierbei der entscheidende Vorteil gegenüber der Auswertung als einzelne unabhängige CRFs. Die durch die Messung gegebenen Informationen werden demnach effektiver genutzt.

Allerdings ist durch dieses Verfahren die Verteilung einiger Zustände der kompletten Sequenz $\mathbf{S}$ nicht eindeutig. Für die Zeitindices, die als unzuverlässig gelten, wurden mehrere Verteilungen bestimmt. Das bedeutet, für $n = 1, 2, \ldots, \tau$ und $l > 1$ gilt, dass $s^{l-1}(T - \tau + n)$ und $s^l(\tau)$ denselben Zustand beschreiben. Dem Prinzip der maximalen Entropie zufolge ist es sinnvoll, die meisten Informationen zur Bestimmung des Zustandes zu nutzen. Dieses gilt es insbesondere zu beachten, wenn eine einzelne Sequenz entsprechend (4.15) bestimmt werden soll. Daher werden die Segmente als CRF ausgewertet, um so beidseitige Informationen zu erhalten, ohne die Möglichkeit der Echtzeit-Bestimmung einzubüßen; das heißt, zu einem vorher fest bestimmten Zeitpunkt findet die Klassifikation eines Zustandes statt, unabhängig von der Länge der Gesamt-Messung. Ein entsprechender Graph befindet sich in Abbildung 4.6. Demnach werden immer nur die neuesten bestimmten Zustände für einen Zeitindex verwendet. Diese enthalten die meiste Information, die lokal konzentriert an einer bestimmten Position im Sequenzgraphen vorhanden ist. Es handelt sich somit um ein für Echtzeitprobleme besser geeignetes Modell als CRFs oder MEMMs.

Eine Interpretation des Modells ist, dass es die Vorteile von MEMMs und CRFs verbindet. Einerseits werden mehr Informationen zur Bestimmung der Zustände verwendet. Deren Aussage ist somit zuverlässiger, was ein Vorteil gegenüber MEMMs ist. Andererseits wird diese Auswertung nur auf Segmenten unter Berücksichtigung historischer Daten betrachtet. Dadurch ist dieses Modell zur Erstellung einer Echtzeitanalyse geeignet, das heißt, es ist auch bei einer kontinuierlichen Aufnahme der Messwerte, zum Beispiel bei einem Überwachungsproblem, möglich, zwischenzeitliche Auswertungen vorzunehmen. Dieses ist ein Vorteil gegenüber CRFs. Die Verzögerung, wann spätestens ein Zustand bestimmt wird, ist höchstens $T - 1$ und hängt demnach von der Segmentlänge ab. Diese Verzögerung ist ein frei wählbarer Parameter und damit abhängig vom behandelten Problem.

Eine weitere Interpretation dieses Modells liefert die Betrachtung von Grenzfällen. Ist $T = 1$ und $\tau = 0$, so ist dieses Modell identisch mit einem MEMM. Ist hingegen $T = N$, das heißt, man betrachtet sämtliche Messwerte für ein Segment, und $\tau = 0$, so entspricht dieses Modell einem linearen CRF. Demnach bietet dieses Modell eine Verbindung zwischen CRFs und MEMMs in einer geschlossenen Form.

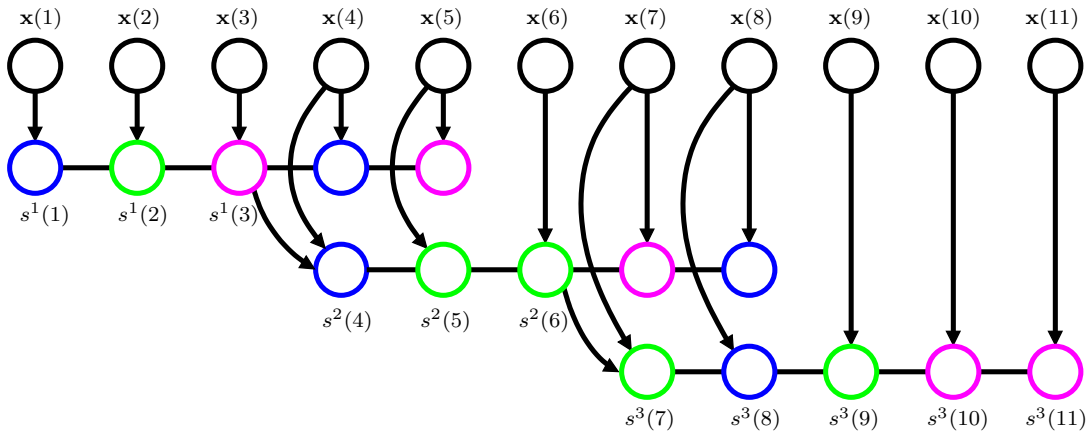


Abbildung 4.6: Graphische Interpretation der sequentiellen Auswertung von Segmenten. Einzelne Segmente (hier der Länge $T = 5$) werden in Form eines CRFs ausgewertet. Informationen dieses Segments werden in das nächste Segment übertragen; insgesamt werden $\tau = 2$ Zustände neu bestimmt. Durch die zusätzlichen Informationen kann die Farbe im nächsten Segment anders sein. Da immer nur linksseitige Informationen übertragen werden, ist dieses Verfahren in einem echtzeitfähigen Algorithmus anwendbar.

Das beschriebene Modell wurde explizit für die Ereignisdetektion entwickelt. Dennoch lässt es sich offensichtlich für jede Form der Zeitreihenanalyse, bei der MEMMs oder lineare CRFs eingesetzt werden, ebenfalls anwenden. Die Menge der Fälle, die sich mit diesem Modell behandeln lassen, ist demnach deutlich größer, was die Relevanz dieses Modells erhöht. Zudem ist es eine verhältnismäßig einfache Erweiterung, die vergleichbar effizient wie CRFs und MEMMs verwendet werden kann.

4.7 Trainingsalgorithmen

Bisher wurde das CRF und das MEMM sowie eine Lösung zwischen diese beiden als Modell vorgestellt. Insbesondere wurden die Auswertung mittels eines CRFs respektive MEMMs besprochen. Ausgeblieben ist bisher das Training dieser Modelle anhand von Beispieldaten. Nach der Modellierung, also die Bestimmung der Potentialfunktion durch $\Phi(C, \mathbf{X})$ in Definition 4.2, wird das Modell an diese Daten angepasst [28]. das bedeutet, es wird der vektorwertige Parameter λ bestimmt. Diesem Training wird sich im Folgenden gewidmet.

Die üblichen Trainingsalgorithmen für CRFs und MEMMs [28, 41, 54, 74] sind überwachte Lernalgorithmen. Das bedeutet, dass neben den Messwerten auch die Färbungen

des zugehörigen Markov-Feldes bekannt sind und im Training verwendet werden. In Abschnitt 4.7.2 wird hingegen eine Erweiterung besprochen, die ein blindes Erlernen ermöglicht, also ein Training, bei dem nur die Messwerte, nicht die zugehörigen Zufallsfelder, bekannt sind [50]. Diese Erweiterung ist für die Ereignisdetektion wichtig, da dieser Fall hier auch vorkommen kann [44]. Dabei wird der Normalfall tiefergehend strukturiert, was zur Entscheidung, ob eine Beobachtung zum Ereignis oder Normalfall gehört, nützlich sein kann.

Im Folgenden wird zunächst ein Algorithmus zum Training der Modelle besprochen, der ebenfalls überwacht ist [28]. Er bildet die Grundlage für den unüberwachten Trainingsalgorithmus und ist damit wichtig für die später folgenden Methoden.

Betrachtet wird zunächst das Training eines allgemeinen CRFs, das heißt ein CRF mit einem allgemeinen, nicht notwendigerweise linearen Zustandsgraphen. Zum Training der Modelle können mehrere unterschiedliche Messungen und dazugehörige Färbungen verwendet werden. Zum Beispiel können mehrere Messreihen unterschiedliche Interpretationen besitzen, in etwa die Segmente in Abschnitt 4.6. Diese Messungen seien $\mathbf{X}^2$, $\mathbf{X}^3, \ldots, \mathbf{X}^M$ und dazugehörige gefärbte Instanzen S^1, S^2, $\ldots S^M$ des Zufallsfeldes S. Gesucht ist der Parameter λ. Für ein trainiertes CRF sollte gelten, dass man für eine Messung $\mathbf{X}^l$ ebenfalls die Färbung S^l erhält,

$$E_S\left\{S|\mathbf{X}^l\right\} = S^l, \tag{4.28}$$

wobei E_S der bedingte Erwartungswert des Zufallsfeldes S bei gegebener Messung $\mathbf{X}^l$ ist. Diese Gleichheit entspricht der Voraussetzung der Verteilung maximaler Entropie, siehe Abschnitt 4.2.

Es wird derjenige Parametervektor gesucht, der die Likelihood für die bekannte Färbung, also $p(S^l|\mathbf{X}^l)$, maximiert. Das Training kann folglich betrachtet werden als ein Optimierungsproblem der Likelihood der Färbungen zu einer gegebenen Messung.

Anstatt die Likelihood direkt zu maximieren, wird diese zunächst logarithmiert:

$$\begin{aligned}\log\left(p(S^l|\mathbf{X}^l)\right) &= \log\left(\frac{1}{Z(\mathbf{X}^l)}\exp\left(\sum_{C\in S^l}\lambda^\top\Phi(C,\mathbf{X}^l)\right)\right)\\ &= \sum_{C\in S^l}\lambda^\top\Phi(C,\mathbf{X}^l) - \log\left(Z(\mathbf{X}^l)\right),\end{aligned} \tag{4.29}$$

wobei die Normalisierungskonstante $Z(\mathbf{X}^l)$ gegeben ist durch

$$Z(\mathbf{X}^l) = \sum_S \exp\left(\sum_{C \in S} \lambda^\top \Phi(C, \mathbf{X}^l)\right)$$

mit der Summation über alle möglichen Färbungen des Graphen. Diese Normalisierungskonstante garantiert, dass $p(S|\mathbf{X}^l)$ eine Wahrscheinlichkeit ist, also dass gilt

$$\sum_S p(S|\mathbf{X}^l) = 1. \tag{4.30}$$

Dieses entspricht der Formulierung der Verteilung maximaler Entropie in Bezug auf die Zustände, siehe hierzu Abschnitt 4.2.

Die Optimierung der logarithmierten Likelihood (4.29) ist eine Maximierung bezüglich des Parameters λ [21, 41]. Der Gradient dieser Funktion nach λ ist

$$\nabla_\lambda \log\left(p(S^l|\mathbf{X}^l)\right) = \sum_{C \in S^l} \Phi(C, \mathbf{X}^l) - \nabla_\lambda \log\left(Z(\mathbf{X}^l)\right), \tag{4.31}$$

wobei für $\nabla_\lambda \log\left(Z(\mathbf{X}^l)\right)$ gilt

$$\begin{aligned} \nabla_\lambda \log\left(Z(\mathbf{X}^l)\right) &= \frac{1}{Z(\mathbf{X}^l)} \sum_S \sum_{C \in S} \exp\left(\lambda^\top \Phi(C, \mathbf{X}^l)\right) \cdot \Phi(C, \mathbf{X}^l) \\ &= E_{\Phi(S,\mathbf{X}^l)}\left\{\Phi(S, \mathbf{X}^l)|\mathbf{X}^l\right\}, \end{aligned} \tag{4.32}$$

$\Phi(S, \mathbf{X}^l) = \sum_{C \in S} \Phi(C, \mathbf{X}^l)$ ist und $E_{\Phi(S,\mathbf{X}^l)}$ der Erwartungswert von $\Phi(S, \mathbf{X}^l)$ und $\mathbf{X}^l$ gegeben ist [28].

Eine Möglichkeit, mittels dieser Gleichungen das CRF zu trainieren, ist eine Maximierung in Richtung des mittleren Gradienten, also ein Gradientenverfahren [21]. Hierfür werden die Gradienten über die Trainingsdatensätze gemittelt. Dadurch ergibt sich der Aktualisierungsschritt für λ.

Das heißt, dass λ zunächst zufällig initiiert wird. Anschließend wird der mittlere Gradient berechnet:

$$\frac{1}{M} \sum_{l=1}^{M} \nabla_\lambda \log\left(p(S^l|\mathbf{X}^l)\right) = \frac{1}{M} \sum_{l=1}^{M} \sum_{C \in S^l} \Phi(C, \mathbf{X}^l) - E_{\Phi(S,\mathbf{X}^l)}\left\{\Phi(S, \mathbf{X}^l)|\mathbf{X}^l\right\}. \tag{4.33}$$

Daraufhin wird λ mit einer zuvor festgelegten Schrittweite aktualisiert:

$$\lambda \leftarrow \lambda + const \cdot \frac{1}{M} \sum_{l=1}^{M} \nabla_\lambda \log \left(p(S^l | \mathbf{X}^l) \right). \tag{4.34}$$

Diese Aktualisierung wird wiederholt, bis die Konvergenz erreicht ist, zum Beispiel, dass die Norm des Gradienten kleiner als ein zuvor festgelegter Schwellwert θ ist:

$$\left\| \frac{1}{M} \sum_{l=1}^{M} \nabla_\lambda \log \left(p(S^l | \mathbf{X}^l) \right) \right\|_2 < \theta. \tag{4.35}$$

Dieses Gradientenverfahren konvergiert, da die Zielfunktion konvex ist [41].

Die Komplexität dieses Trainingsverfahrens resultiert vor allem in der Auswertung des Markov-Zufallsfeldes an sich, vor allem in (4.32). Hierfür muss jede mögliche Färbung berücksichtigt werden. Mit großer Anzahl an möglichen Färbungen kann dieses für allgemeine Markov-Felder sehr aufwendig werden. Wenn zum Beispiel in einem Graphen mit N Knoten jeder einzelne Knoten eine von K Farben annehmen kann, so existieren N^K unterschiedliche Graphen-Färbungen, die berücksichtigt werden müssen.

Lineare CRFs und MEMMs besitzen jedoch den Vorteil, dass diese Färbungen deutlich schneller bestimmt werden können. Da dieses auch der wichtigste Spezialfall für die Ereignisdetektion ist, wird dieser im nächsten Abschnitt noch einmal im Detail erläutert, auch wenn sie allgemein mit dem hier vorgestellten Verfahren trainiert werden können.

4.7.1 Training eines linearen bedingten Markov-Zufallsfeldes

Bisher wurde das Training von allgemeinen CRFs betrachtet. Dabei wurde gezeigt, dass das Training eines CRFs sehr aufwendig sein kann. Es existieren jedoch einige Formen von CRFs, die in praktikabler Geschwindigkeit trainiert werden können. Zu diesen gehört das lineare CRF, das für die Ereignisdetektion auch interessant ist. Zudem ist es die Form, die zur Vorstellung von CRFs verwendet wurde [41] und die in den folgenden Kapiteln insbesondere betrachtet wird.

In diesem Abschnitt wird das überwachte Training eines linearen CRFs diskutiert [28]. Dabei wird auf die spezielle Struktur dieses Modells Rücksicht genommen. Dadurch wird das Training deutlich effizienter, ferner ist diese Form interessant, da hiermit das Markov-Zufallsfeld, in diesem Fall eine Markov-Kette, der linearen Abfolge von Messwerten entsprechen kann. Es ist somit für die Analyse von Zeitreihen ein interessanter Spezialfall, zudem ein verhältnismäßig einfach zu bestimmendes Modell.

Es seien $\mathbf{X}^2$, $\mathbf{X}^3$, …, $\mathbf{X}^M$ Messungen und $\mathbf{S}^1$, $\mathbf{S}^2$, …, $\mathbf{S}^M$ die Färbungen der zugehörigen Zustandssequenzen, also Instanzen der Markov-Kette. Dann folgt aus der Definition für lineare CRFs (Definition 4.3), dass sich die logarithmierte Likelihood der Zustandssequenz $\mathbf{S}^l$ ausdrücken lässt durch

$$\begin{aligned}\log\left(p(\mathbf{S}^l|\mathbf{X}^l)\right) &= \log\left(\frac{1}{Z(\mathbf{X}^l)}\exp\left(\sum_{n=1}^{N_l}\lambda^\top\Phi(s^l(n-1),s^l(n),\mathbf{X}^l,n)\right)\right)\\ &= \sum_{n=1}^{N_l}\lambda^\top\Phi(s^l(n-1),s^l(n),\mathbf{X}^l,n) - \log\left(Z(\mathbf{X}^l)\right), \qquad (4.36)\end{aligned}$$

wobei N_l die Länge der Zustandssequenz $\mathbf{S}_l$ ist, $s^l(n-1)$ und $s^l(n)$ Zustände dieser Sequenz zum Zeitindex $n-1$ respektive n sind. Die Normalisierungskonstante $Z(\mathbf{X}^l)$ ist hierbei gegeben durch

$$Z(\mathbf{X}^l) = \sum_{\mathbf{S}}\exp\left(\sum_{n=1}^{N_l}\lambda^\top\Phi(s(n-1),s(n),\mathbf{X}^l,n)\right).$$

Dies ist die Normalisierung über alle möglichen Sequenzfärbungen derselben Länge wie $\mathbf{S}^l$ unter der Berücksichtigung der Messungen $\mathbf{X}^l$. Diese Umformung der Summe über die Cliquen hin zu einer Summe über den Zeitindex ergibt sich durch den Umstand, dass in einer linearen Kette jeweils zwei benachbarte Knoten eine maximale Clique bilden, das heißt, jeder Knoten des von ihnen gebildeten Subgraphen ist mit jedem anderen verbunden und es gibt keinen Knoten des Graphen, der mit jedem Knoten dieses Subgraphen adjazent ist. Aus diesem Umstand ist das lineare CRF effizient zu berechnen.

Der Gradient der logarithmierten Likelihood ist

$$\nabla_\lambda\log\left(p(\mathbf{S}^l|\mathbf{X}^l)\right) = \sum_{n=1}^{N_l}\lambda^\top\Phi(s^l(n-1),s^l(n),\mathbf{X},n) - E_{\Phi(\mathbf{S},\mathbf{X}^l)}\left\{\Phi(\mathbf{S},\mathbf{X}^l)|\mathbf{X}^l\right\}, \qquad (4.37)$$

mit $\Phi(\mathbf{S},\mathbf{X}^l) = \sum_{n=1}^{N_l}\Phi(s(n-1),s(n),\mathbf{X}^l,n)$.

Der Erwartungswert $E_{\Phi(\mathbf{S},\mathbf{X}^l)}\left\{\Phi(\mathbf{S},\mathbf{X}^l)|\mathbf{X}^l\right\}$ von $\Phi(\mathbf{S},\mathbf{X}^l)$ lässt sich effizient ausrechnen [28]. Hierfür werden, wie bei der Auswertung, beide Pfade berücksichtigt

sowie die Wahrscheinlichkeit jedes Übergangs. Dieses erfolgt aus der Bestimmung des Erwartungswertes: für diesen gilt

$$E_{\Phi(\mathbf{S},\mathbf{X}^l)}\left\{\Phi(\mathbf{S},\mathbf{X}^l)|\mathbf{X}^l\right\} = E_{\Phi(\mathbf{S},\mathbf{X}^l)}\left\{\sum_{n=1}^{N_l}\Phi(s(n-1),s(n),\mathbf{X}^l,n)|\mathbf{X}^l\right\}$$
$$= \sum_{n=1}^{N_l} E_{\Phi(\mathbf{S},\mathbf{X}^l)}\left\{\Phi(s(n-1),s(n),\mathbf{X}^l,n)|\mathbf{X}^l\right\}. \qquad (4.38)$$

Zur Bestimmung des Erwartungswertes von Φ über die gesamte Sequenz genügt es folglich, nur die einzelnen Übergänge, das sind die Cliquen, zu betrachten. Die nichtnormalisierte Wahrscheinlichkeit für einen speziellen Übergang von der Farbe ζ_j in die Farbe ζ_k zum Zeitpunkt n ist gleich der Wahrscheinlichkeit, Farbe ζ_j zum Zeitpunkt $n-1$ zu beobachten, multipliziert mit der Wahrscheinlichkeit, zum Zeitpunkt n die Farbe ζ_k zu beobachten, multipliziert mit der Wahrscheinlichkeit des Übergangs von ζ_j zu ζ_k. Da jeder Knoten und jede Kante den Graphen teilt, das heißt, dass das Markov-Feld ohne diesen Knoten respektive diese Kante in zwei statistisch unabhängige Zufallsfelder zerfällt, müssen zur Bestimmung der Wahrscheinlichkeit von ζ_j zum Zeitpunkt n in der Kette $\mathbf{S}^l$ nur die Zustände $s^l(1)$, $s^l(2)$, …, $s^l(n-1)$ betrachtet werden, für die Wahrscheinlichkeit von Zustand ζ_j zum Zeitpunkt n die Zustände $s^l(n+1)$, $s^l(n+2)$, …, $s^l(N_l)$.

Sei deswegen

$$\tilde{\alpha}^l(n,k) = \begin{cases} \exp\left(\lambda^\top \Phi(s_0,\zeta_k,\mathbf{X}^l,1)\right), & \text{falls } n=1 \\ \sum_{j=1}^{K} \alpha^l(n-1,j)\cdot\exp\left(\lambda^\top\Phi(\zeta_j,\zeta_k,\mathbf{X}^l,n)\right), & \text{sonst} \end{cases}$$
$$\alpha^l(n,k) = \frac{\tilde{\alpha}^l(n,k)}{\sum_{j=1}^{K}\tilde{\alpha}^l(n,j)} \qquad (4.39)$$

der vorwärts gerichtete Pfad und

$$\tilde{\beta}^l(n,k) = \begin{cases} \exp\left(\lambda^\top \Phi(\zeta_k,s_0,\mathbf{X}^l,N+1)\right) & \text{falls } n=T \\ \sum_{j=1}^{K} \beta^l(n+1,j)\cdot\exp\left(\lambda^\top\Phi(\zeta_k,\zeta_j,\mathbf{X}^l,n+1)\right) & \text{sonst} \end{cases}$$
$$\beta^l(n,k) = \frac{\tilde{\beta}^l(n,k)}{\sum_{j=1}^{K}\tilde{\beta}^l(n,k)}, \qquad (4.40)$$

der rückwärts gerichtete Pfad bezüglich der Messwerte $\mathbf{X}^l$. Diese übertragen die Infor-

mationen zu der Wahrscheinlichkeit jedes Knotens in die entsprechende Richtung. Sei ferner $Q^l(n, j, k)$ definiert durch

$$Q^l(n, j, k) = \exp\left(\lambda^\top \Phi(\zeta_j, \zeta_k, \mathbf{X}^l, n)\right), \tag{4.41}$$

dann ist die Wahrscheinlichkeit für den Übergang von Zustand ζ_j zu Zustand ζ_k zum Zeitpunkt n in der gesamten Zustandssequenz unter Berücksichtigung der Messsequenz $\mathbf{X}^l$ [28], also $P(s(n-1) = \zeta_j, s(n) = \zeta_k|\mathbf{X}^l)$, gegeben durch

$$\begin{aligned}
\tilde{P}(s^l(n-1) = \zeta_j, s^l(n) = \zeta_k|\mathbf{X}^l) &= \alpha^l(n-1, j) \cdot Q^l(n, j, k) \cdot \beta^l(n, k) \\
P(s^l(n-1) = \zeta_j, s^l(n) = \zeta_k|\mathbf{X}^l) &= \frac{\tilde{P}(s^l(n-1) = \zeta_j, s^l(n) = \zeta_k|\mathbf{X}^l)}{\sum\limits_{s_1,s_2} \tilde{P}(s^l(n-1) = s_1, s^l(n) = s_2|\mathbf{X}^l)},
\end{aligned} \tag{4.42}$$

wobei s_1 und s_2 über alle möglichen Färbungen eines Zustands laufen, also $s_1, s_2 \in \{\zeta_1, \zeta_2, \ldots, \zeta_k\}$. Dementsprechend ist der Erwartungswert für die Merkmalsfunktion der Potentialfunktion, also Φ, zu einem einzelnen Zeitpunkt gegeben durch

$$\begin{aligned}
E_{\Phi(\mathbf{S}^l,\mathbf{X}^l)}\left\{\Phi(s(n-1), s(n), \mathbf{X}^l, n)|\mathbf{X}^l\right\} = \\
\sum_{j,k=1}^{K} P\left(s^l(n-1) = \zeta_j, s^l(n) = \zeta_k|\mathbf{X}^l\right) \cdot \Phi(\zeta_j, \zeta_k, \mathbf{X}^l, n)
\end{aligned} \tag{4.43}$$

und somit für die Summe in (4.38)

$$\begin{aligned}
E_{\Phi(\mathbf{S}^l,\mathbf{X}^l)}\left\{\Phi(\mathbf{S}^l, \mathbf{X}^l)|\mathbf{X}^l\right\} = \\
\sum_{n=1}^{N_l} \sum_{j,k=1}^{K} P\left(s^l(n-1) = \zeta_j, s^l(n) = \zeta_k|\mathbf{X}^l\right) \cdot \Phi(\zeta_j, \zeta_k, \mathbf{X}^l, n).
\end{aligned} \tag{4.44}$$

Die Berechnung dieses Erwartungswertes ist demnach deutlich effizienter als in allgemeinen CRFs. In einem allgemeinen CRF sind N^K Färbungen des Graphen möglich. Bei linearen CRFs müssen für die Wege α und β in jedem Schritt K Färbungen betrachtet werden. Für den Übergang Q sind noch einmal K^2 Färbungen zu betrachten. Insgesamt werden demnach $N_l \cdot (2K + K^2)$ Färbungen einzelner Zustände für die Auswertung dieses Erwartungswertes benötigt, was für praktische Längen der Sequenzen N_l und Farben K eine deutlich effizientere Auswertung ist.

Mit $E_{\Phi(\mathbf{S}^l,\mathbf{X}^l)}\left\{\Phi(\mathbf{S}^l, \mathbf{X}^l)|\mathbf{X}^l\right\}$ kann demnach ein CRF trainiert werden, zum Beispiel mittels des Gradientenverfahrens [21]. Dafür wird der Vektor λ zufällig initiiert. In den

Iterationsschritten des Trainings wird der mittlere Gradient wie in (4.33) berechnet, das ist

$$\frac{1}{M}\sum_{l=1}^{M}\nabla_\lambda \log\left(p(\mathbf{S}^l|\mathbf{X}^l)\right) = \\ \frac{1}{M}\sum_{l=1}^{M}\sum_{n=1}^{N_l}\Phi(s^l(n-1), s^l(n), \mathbf{X}^l, n) - E_{\Phi(\mathbf{S}^l,\mathbf{X}^l)}\left\{\Phi(\mathbf{S}^l, \mathbf{X}^l)|\mathbf{X}^l\right\}. \quad (4.45)$$

In das Training wird für gewöhnlich ein Strafterm hinzugefügt [28]. Der Grund ist, dass sowohl MEMMs als auch CRFs zur Überanpassung neigen. Das heißt für CRFs und MEMMs, dass die Beträge der Gewichte in λ zu groß werden können. Dieses lässt sich direkt mit dem Einfluss einer Dimension der Messung auf das Ergebnis gleich setzen, wie aus der Definition von CRFs folgt. Der Strafterm, der für gewöhnlich verwendet wird, ist $\frac{1}{2\sigma}\lambda$, wobei σ ein frei zu wählender Parameter ist. Der Aktualisierungsschritt ist demnach

$$\lambda \leftarrow \lambda + const \cdot \left(\frac{1}{M}\sum_{l=1}^{M}\nabla_\lambda \log\left(p(\mathbf{S}^l|\mathbf{X}^l)\right) - \frac{1}{2\sigma}\lambda\right), \quad (4.46)$$

wobei das Training abgeschlossen ist, wenn für einen vorher festgelegten Schwellwert θ gilt

$$\left\|\frac{1}{M}\sum_{l=1}^{M}\nabla_\lambda \log\left(p(\mathbf{S}^l|\mathbf{X}^l)\right) - \frac{1}{2\sigma}\lambda\right\|_2 < \theta. \quad (4.47)$$

Da die Zielfunktion streng konvex ist [28, 41], konvergiert dieses Verfahren zu einem eindeutigen Maximum.

Neben diesem Verfahren existieren welche nach der generalisierten iterativen Skalierung [41] sowie Optimierungsverfahren höherer Ordnung, zum Beispiel Quasi-Newton-Verfahren wie das L-BFGS-Verfahren [21, 28]. Der hier diskutierte Trainingsalgorithmus bildet jedoch die Basis für das blinde Training in Abschnitt 4.7.2. Dieses Verfahren basiert auf einem Wechsel zwischen der Schätzung einer Sequenz und der Anpassung an diese, entsprechend einem EM-Algorithmus [56]. Da in diesem Fall eine Überanpassung an eine temporäre Zustandssequenz verhindert werden soll, wurde auf Verfahren höherer Ordnung verzichtet.

4.7.2 Blindes Training

Das Training für CRFs und MEMMs wurde bisher nur anhand von bekannten Zustandssequenzen zu dazugehörigen Messungen besprochen. Folglich handelt es sich bei diesen Algorithmen um überwachtes Training, bei denen die Antwort des Modells auf eine Messung $\mathbf{X}^l$ an $\mathbf{S}^l$ angepasst wird. Damit das Modell entsprechend trainiert werden kann, müssen in den regulären Trainingsalgorithmen alle Zustände in den Trainingsdaten repräsentiert werden. Dadurch erlernt das CRF die entsprechenden Ausprägungen für jeden Zustand.

Für die Ereignisdetektion ist dieses Vorgehen nicht immer geeignet. Einerseits sind die Ereignisse, die in dieser Arbeit vornehmlich besprochen werden, nicht bekannt und dadurch nicht in der Trainingsmenge repräsentiert. Dieses Problem wird in Kapitel 5 behandelt. Es bildet eine deutliche Erweiterung klassischer Verfahren und ist für die Problemstellung dieser Arbeit spezialisiert.

Andererseits ist die in dieser Arbeit besprochene Aufgabenstellung der Ereignisdetektion, dass das Modell anhand von Daten trainiert werden soll, von denen nur bekannt ist, dass sie dem Normalfall angehören. Das heißt, dass zunächst zwei Fälle - das Ereignis und der Normalfall - auseinander gehalten werden, und nur von letzterem existieren Trainingsbeispiele. Die üblichen Trainingsalgorithmen sind somit ungeeignet, diesen Fall zu behandeln.

Zur Analyse des Normalfalls kann es hilfreich sein, ihn in mehrere Unterfälle zu untergliedern. Damit können auch mehrere Farben den Normalfall beschreiben; jede Farbe beschreibt eine von K speziellen Ausprägungen. Allerdings ist für diese Aufteilung des Normalfalls nicht die Einteilung bekannt. Sinnvoll wäre es, wenn beim Training des CRFs diese Ausprägungen ohne manuellen Einfluss erlernt werden können. Dadurch kann das CRF im Training selbst den Normalfall in K Fälle aufteilen. Das CRF respektive MEMM erlernt somit die Zustände blind, darum handelt es sich um ein unüberwachtes Trainingsverfahren [50].

Das in diesem Abschnitt diskutierte Trainingsverfahren basiert auf dem Prinzip des EM-Algorithmus [56]. Dabei werden im sogenannten E-Schritt (kurz für englisch: *Expectation*) zu den Sequenzen an Messungen mittels des CRFs Sequenzen an Zuständen geschätzt. In dem darauf folgenden M-Schritt (kurz für englisch: *Maximization*) wird das CRF an diese geschätzten Sequenzen angepasst. Lafferty et. al. nahmen an, dass eine solche Methode möglich ist, um ein CRF blind zu trainieren [41]. Jedoch erläuterten sie einen solchen Algorithmus nicht im Detail. Wie sich in Experimenten herausstellte, sind deutliche Änderungen am Trainingsalgorithmus notwendig, damit dieses blinde

Training angewendet werden kann. Diese Änderungen wurden im Rahmen dieser Arbeit entwickelt und werden im folgenden erläutert.

Das Verfahren funktioniert prinzipiell mit einer beliebigen Anzahl von Trainingsbeispielen. Das heißt, es können mehrere Sequenzen unterschiedlicher Länge behandelt werden. Aus Gründen der Übersichtlichkeit wird hier jedoch angenommen, dass nur eine solche Sequenz existiert.

Gegeben sei nun die Trainingssequenz $\mathbf{X}$ sowie die Anzahl der erwünschten Zustände K. Ferner sei weiterhin die Funktion $\Phi(s_1, s_2, \mathbf{X}, n)$ gegeben. Mittels dieser Trainingsbeispiele und der Funktion soll ein CRF trainiert werden, das zu jedem Datenwert die Wahrscheinlichkeit für einen selbst geschätzten Zustand angibt.

4.7.2.1 Schätzung einer Zustandssequenz

Zur Initialisierung des Trainings wird zunächst λ zufällig bestimmt. Prinzipiell ließe sich hiermit eine Sequenz schätzen nach Gleichung (4.15). Dieses ist das Verfahren für den E-Schritt, das ursprünglich vorgeschlagen wurde [41]. Dennoch erfüllt diese Sequenz nicht immer die erwünschten Voraussetzungen.

Vor allem ist es wünschenswert, dass die geschätzte Sequenz alle K Färbungen besitzt. Dieses ist ein vorher festgesetzter Parameter, der auch ausgeschöpft werden soll. Durch ihn kann die Komplexität des Modells vor dem Training festgesetzt werden. Zudem wird die maximal mögliche Informationsnutzung über die Messwerte ermöglicht.

Ferner kann es von Vorteil sein, wenn sich die Information nicht auf wenige Farben konzentriert, also im Mittel in der geschätzten Sequenz einige Farben deutlich häufiger auftreten als andere. Dadurch können mehrere unterschiedliche Ausprägungen des Normalfalls auch mit kleineren Unterschieden identifiziert werden. Spezialisierungen, also Anpassungen auf sehr prägnante Merkmale, sollten dennoch nicht ausgeschlossen werden. Beide Eigenschaften, die Spezialisierung und die Generalisierung, müssen demnach in der Sequenz berücksichtigt werden. Da die Information über diese Ausprägungen zunächst nicht bekannt ist, sollten vor dem Training die Färbungen möglichst gleich auf die Sequenz verteilt werden. Die Information über besonders prägnante Zustände wird beim Training selbst erworben.

Eine Möglichkeit, diese Punkte zu vermitteln, ist mittels der Maximierung der Entropie in dem Graphen möglich [10, 18, 24, 67, 81]. Im Fall einer Markov-Kette kann von der Maximierung der *Sequenzentropie* gesprochen werden. Das bedeutet, dass erwartet wird,

dass jeder Zustand in der geschätzten Sequenz gleich häufig beobachtet wird. Dabei wird die Sequenzentropie $H_S(\mathbf{S})$ definiert durch

$$H_S(\mathbf{S}) = -\frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{K} P(s(n) = \zeta_k|\mathbf{X}) \cdot \log\left(P(s(n) = \zeta_k|\mathbf{X})\right), \tag{4.48}$$

wobei N die Länge der Sequenz ist.

Diese Definition der Sequenzentropie ist im Fall des CRFs differenzierbar, wie sich aus den diskutierten Trainingsalgorithmen ersehen lässt. Sie lässt sich somit in das Gradientenverfahren integrieren, um den Gewichtsvektor λ zu bestimmen.

Dabei ergibt sich jedoch ein Problem mit diesem Ansatz. Das Training eines CRFs lässt sich derart interpretieren, dass die Beschreibung eines Zustandes im Zusammenhang seiner Nachbarn angepasst wird. Das bedeutet, dass die Entropie zu einem spezifischen Zeitpunkt $h_S(n, \mathbf{S})$ mit

$$h_S(n, \mathbf{S}) = -\sum_{k=1}^{K} P(s(n) = \zeta_k|\mathbf{X}) \cdot \log\left(P(s(n) = \zeta_k|\mathbf{X})\right) \tag{4.49}$$

minimiert wird, was dem Prinzip der Entropiemaximierung widerspricht. Ein Verfahren, das beides zugleich berücksichtigt, ist daher nicht notwendig effizient, das heißt, nicht jeder Iterationsschritt des Trainingsverfahrens führt zu einem besseren Ergebnis mit einer besseren Identifikation des Zustandes bei gleichzeitiger Zunahme der Entropie in einer geschätzten Sequenz.

Das eigentlich zu verfolgende Ziel ist es, nicht die Sequenzentropie zu erhöhen, sondern die Entropie nach der Schätzung der Zustände, das heißt die *Zustandsentropie* $H_{\hat{S}}(\hat{\mathbf{S}})$ mit

$$\begin{aligned} H_{\hat{S}}(\hat{\mathbf{S}}) &= \sum_{k=1}^{K} h_{\hat{S}}(k, \hat{\mathbf{S}}) \\ h_{\hat{S}}(k, \hat{\mathbf{S}}) &= -\left(\frac{1}{N}\sum_{n=1}^{N}[\![\hat{s}(n) = \zeta_k]\!]\right) \cdot \log\left(\frac{1}{N}\sum_{n=1}^{N}[\![\hat{s}(n) = \zeta_k]\!]\right) \\ \hat{s}(n) &= \arg\max_{\zeta} P(s(n) = \zeta|\mathbf{X}). \end{aligned} \tag{4.50}$$

Diese Entropie ist definiert über die Auftrittshäufigkeit der Zustände, nicht über die Sequenz. Eine Erhöhung der Entropie zu einem Zeitpunkt ist somit hier nicht enthalten.

Eine Reduzierung der Entropie wie sie im Training des CRFs auftritt kann somit zu jedem Zeitpunkt akzeptiert werden.

Mit diesem Prinzip ist die Auftrittshäufigkeit eines Zustandes in der geschätzten Sequenz zu maximieren, nicht die Sequenzentropie selbst. Diese Funktion ist nicht differenzierbar, da durch die Nutzung der Zustände mit der maximalen Likelihood eine Quantisierung der Wahrscheinlichkeit impliziert wird. Stattdessen kann (4.50) als Gütemaß für eine Zustandssequenz genutzt werden. An diese wird in dem Training das CRF angepasst. Der M-Schritt wird im Vergleich zu herkömmlichen Trainingsverfahren nicht verändert, ausschließlich der E-Schritt wird für das gegebene Problem angepasst.

Hierfür wird zunächst der Begriff der *optimalen Markov-Kette* definiert. Diese Definition ermöglicht es, Forderungen an ein CRF zu stellen, dass es sowohl im Training konvergiert als auch die Zustandsentropie erhöht.

Definition 4.4 (Optimale Markov-Kette) *Sei* **S** *eine Markov-Kette der Länge N, seien $\zeta_1, \zeta_2, \ldots, \zeta_K$ mögliche Farben, sodass für jeden Knoten der Markov-Kette gilt $s(n) \in \{\zeta_1, \zeta_2, \ldots, \zeta_K\}$. Die Markov-Kette heißt* optimal*, wenn für ihre Verteilung gilt, dass erwartet wird, dass jede Farbe gleich häufig auftritt, das heißt*

$$E\left\{\sum_{n=1}^{N}[\![s(n)=\zeta_1]\!]\right\} = E\left\{\sum_{n=1}^{N}[\![s(n)=\zeta_2]\!]\right\} = \cdots = E\left\{\sum_{n=1}^{N}[\![s(n)=\zeta_K]\!]\right\}. \tag{4.51}$$

Insbesondere gilt für jedes $k = 1, 2, \ldots, K$, dass der Erwartungswert über die Auftrittshäufigkeit einer Farbe $E\left\{\sum_{n=1}^{N}[\![s(n)=\zeta_1]\!]\right\} = \frac{N}{K}$ ist.

Auf die exakte Gleichheit der Auftrittshäufigkeit in einer Sequenz,

$$\sum_{n=1}^{N}[\![s(n)=\zeta_1]\!] = \sum_{n=1}^{N}[\![s(n)=\zeta_2]\!] = \cdots = \sum_{n=1}^{N}[\![s(n)=\zeta_K]\!],$$

wurde verzichtet, da dafür notwendigerweise gelten müsste, dass $N = m \cdot K$ für ein $m \in \mathbb{N}$ gilt. Offensichtlich ist das nicht notwendigerweise der Fall. Stattdessen wird die schwächere Form des Erwartungswertes für diese Definition gewählt. Die Markov-Kette ist in der Form optimal, dass sie die höchste Entropie ermöglicht, das heißt, die Informationen in der Kette bestmöglich nach dem Prinzip der maximalen Entropie verteilt sind.

Das Ziel ist es somit, für die Schätzung einer Sequenz im E-Schritt eine optimale Markov-Kette zu erzeugen. Da im Training das CRF an eine Zustandssequenz angepasst wird, also eine Instanz der Markov-Kette, genügt dieses Vorgehen. An diese wird das CRF angepasst, dadurch wird das Zufallsfeld des CRFs zu einer optimalen Markov-Kette. Der Einfachheit halber wird die Sequenz, die mittels der optimalen Markov-Kette geschätzt wird, *optimale Sequenz* genannt. Da nicht zwingend jeder Zustand gleich häufig auftritt, da andernfalls die Gleichheit der Auftrittshäufigkeit der Farben angenommen werden würde, ist eine optimale Sequenz nicht eindeutig. Daher muss die Bestimmung der optimalen Sequenz in jedem E-Schritt erneut erfolgen.

Im Folgenden wird in einem Satz gezeigt, wie eine optimale Sequenz mittels eines CRFs erzeugt werden kann, auch wenn die wahrscheinlichste Kette des CRFs selbst nicht optimal ist.

Satz 4.1 (Optimale Sequenz.) *Sei λ der Gewichtsvektor eines CRFs und $\Phi(s_1, s_2, \mathbf{X}, n)$ die das CRF definierende Funktion, mit $s_1, s_2 \in \{\zeta_1, \zeta_2, \ldots, \zeta_K\}$. Dann ist $\hat{\mathbf{S}}$ mit $\hat{\mathbf{S}} = \hat{s}(1), \hat{s}(2), \ldots, \hat{s}(N)$,*

$$\hat{s}(n) = \arg\max_{\zeta} \frac{P(s(n) = \zeta | s(n-1), s(n+1), \mathbf{X})}{\sum_m P(s(m) = \zeta | s(m-1), s(m+1), \mathbf{X})} \tag{4.52}$$

mit $\zeta \in \{\zeta_1, \zeta_2, \ldots, \zeta_K\}$ optimal.

Beweis. Sei

$$P_k(n) = \frac{P(s(n) = \zeta_k | s(n-1), s(n+1), \mathbf{X})}{\sum_m P(s(m) = \zeta_k | s(m-1), s(m+1), \mathbf{X})}$$

entsprechend der Voraussetzungen des Satzes. Dann gilt nach Definition $\sum_{n=1}^{N} P_k(n) = 1$ und $0 < P_k(n)$ nach der Definition eines CRFs. Insbesondere gilt

$$\sum_{k=1}^{K} \frac{P_k(n)}{\sum_{l=1}^{K} P_l(n)} = 1$$

und daher ist $P_k(n) \cdot \left(\sum_{l=1}^{K} P_l(n) \right)^{-1}$ eine Wahrscheinlichkeit in k. Weil $\sum_{l=1}^{K} P_l(n)$ konstant ist bezüglich k, gilt

$$\sum_{n=1}^{N} \frac{P_k(n)}{\sum_{l=1}^{K} P_l(n)} = C, \tag{4.53}$$

das bedeutet, die Summe über alle $P_k(n)$ über n ist für alle Farben gleich. Da für die Summe über alle Farben gilt $\sum_{k=1}^{K} P_k(n) \cdot \left(\sum_{l=1}^{K} P_l(n) \right)^{-1} = 1$, gilt ebenfalls

$$N = \sum_{n=1}^{N} \sum_{k=1}^{K} \frac{P_k(n)}{\sum_{l=1}^{K} P_l(n)} = \sum_{k=1}^{K} \sum_{n=1}^{N} \frac{P_k(n)}{\sum_{l=1}^{K} P_l(n)} \overset{(4.53)}{=} K \cdot C, \tag{4.54}$$

und aus diesem Grund ist $C = N \cdot K^{-1}$. Dieses ist der Erwartungswert der Auftrittswahrscheinlichkeit einer Farbe. Deshalb erfüllt $P_k(n) \cdot \left(\sum_{l=1}^{K} P_l(n) \right)^{-1}$ die Bedingungen einer optimalen Markov-Kette, siehe Definition 4.4. Weil $\sum_{l=1}^{K} P_l(n)$ konstant in Bezug auf k ist, gilt

$$\arg \max_k P_k(n) = \arg \max_k \frac{P_k(n)}{\sum_{l=1}^{K} P_l(n)},$$

und darum ist $\hat{\mathbf{S}}$ mit $\hat{s}(n))$ geschätzt wie in (4.52) eine optimale Sequenz.

qed.

Mit dieser optimalen Sequenz existiert somit eine Möglichkeit, eine Sequenz zu erzeugen, an die das CRF angepasst werden kann. Die Verwendung einer optimalen Sequenz sorgt dafür, dass zum Training alle möglichen Farben genutzt werden. Sie verhindert andererseits nicht, dass besonders prägnante, das heißt von übrigen Messungen stark abweichende, Unterklassen der Messwerte ebenfalls berücksichtigt werden, selbst wenn diese seltener als $N \cdot K^{-1}$ sind. Daher ist dieses ein praktisches Vorgehen, um im E-Schritt des unüberwachten Trainings eines CRFs eine Sequenz zu schätzen.

4.7.2.2 Maximierung der Likelihood

Im letzten Abschnitt wurde eine optimale Sequenz geschätzt, an die ein CRF angepasst werden soll [49, 50]. Dieses entspricht dem E-Schritt in einem EM-Algorithmus [56]. Der zweite Schritt ist der M-Schritt. Eine entsprechende Anpassung des Modells wird in diesem Abschnitt besprochen. Die Basis für diesen Schritt bildet die Optimierung der logarithmierten Likelihood nach Gleichung (4.46).

Sei $\hat{\mathbf{S}}$ eine optimale Sequenz für ein lineares CRF mit Gewichtsvektor λ. Dieser Gewichtsvektor definiert das aktuelle CRF, mit Hilfe der optimalen Sequenz wird ein neues CRF durch die Aktualisierung des Gewichtsvektors λ nach

$$\lambda \leftarrow \lambda + const \cdot \left(\nabla_\lambda \log \left(p(\mathbf{S}|\mathbf{X}) \right) - \frac{1}{2\sigma} \lambda \right) \tag{4.55}$$

bestimmt. Dieses führt zu einem M-Schritt für einen Trainings-Algorithmus, der einem EM-Algorithmus entspricht. Der Optimierungsschritt ist effizient, das heißt, nach einem Optimierungsschritt hat die geschätzte Markov-Kette eine höhere Entropie und gleichzeitig für jeden Zeitpunkt eine genauere Identifikation des trainierten Zustandes: Ist das Markov-Modell, das durch das CRF erzeugt wird, optimal, so beeinflusst die Normierung zur Erzeugung einer optimalen Sequenz nicht die Schätzung dieser Sequenz, da die Auftrittshäufigkeiten bereits gleich sind; das bedeutet, durch die Anpassung an diese optimale Sequenz, dass der Trainingsalgorithmus konvergiert. Gleichzeitig ist im M-Schritt eine streng Funktion [28, 41], wodurch nach diesem Schritt die Likelihood der im E-Schritt geschätzten optimalen Sequenz höher ist.

Jedoch lässt sich dieses einfache Vorgehen durch simple Modifikationen in speziellen, häufig vorkommenden Fällen optimieren. Unter bestimmten Umständen lässt sich ein Gewichtsvektor λ derart kontrollieren, dass die erzeugte Sequenz immer eine hohe Entropie besitzt. Dieses bedeutet eine geschickte Wahl eines Startwertes und zugleich eine effiziente Schätzung eines neuen Gewichtsvektors. Somit kann der Algorithmus, sofern die Daten die entsprechenden Eigenschaften besitzen, in weniger Optimierungsschritten konvergieren.

Anpassungen des Trainingsverfahrens für Spezialfälle. Da für die Potentialfunktion die Funktion $\Phi(s_1, s_2, \mathbf{X}, n)$ essentiell ist, ist es auch notwendig, diese für die Optimierung des Lernalgorithmus zu berücksichtigen. Für die effizienten Schritte wird zunächst eine Funktion vorgestellt, die oft verwendet wird, zum Beispiel in [28], und auch die Grundlage für die Ereignisdetektion im nächsten Kapitel bietet. Die Funktion soll, wie folgend besprochen, separierbar sein. Diese Eigenschaft ist nicht notwendig für CRFs, dennoch werden häufig derartige Merkmalsfunktionen Φ verwendet [28], die diese Eigenschaft besitzen, selbst wenn sie nicht genutzt wird. Darum ist dieses Vorgehen praktisch relevant.

Es sei $\Phi(s_1, s_2, \mathbf{X}, n)$ definiert durch

$$\Phi(s_1, s_2, \mathbf{X}, n) = \begin{bmatrix} [[\![s_2 = \zeta_k]\!] \cdot \mathbf{x}(n)]_{k=1}^{K} \\ \left[[[\![s_1 = \zeta_j]\!] \cdot [\![s_2 = \zeta_k]\!]]_{k=1}^{K}\right]_{j=1}^{K} \end{bmatrix}. \tag{4.56}$$

Die erste Zeile in (4.56) beschreibt den Zusammenhang der Messwerte mit dem CRF, also die Informationsextraktion. Die zweite Zeile beschreibt die Markov-Kette und damit den Zusammenhang zwischen den Zuständen. Die Gewichte für diese vektorwertige Funktion werden im Training bestimmt. Diese lässt sich zerlegen in $K + K^2$ Funktionen mit

$$\begin{aligned} \Phi_k(s_2, \mathbf{X}, n) &= [\![s_2 = \zeta_k]\!] \cdot \mathbf{x}(n) \\ \Phi_{kj}(s_1, s_2) &= [\![s_1 = \zeta_j]\!] \cdot [\![s_2 = \zeta_k]\!], \end{aligned}$$

und damit gilt, dass sich (4.56) darstellen lässt durch

$$\Phi(s_1, s_2, \mathbf{X}, n) = \begin{bmatrix} [\Phi_k(s_2, \mathbf{X}, n)]_{k=1}^{K} \\ \left[[\Phi_{kj}(s_1, s_2)]_{k=1}^{K}\right]_{j=1}^{K} \end{bmatrix}.$$

Diese Zerlegung der Funktion $\Phi(s_1, s_2, \mathbf{X}, n)$ erlaubt eine entsprechende Zerlegung des Gewichtsvektors, die zur Beschreibung des unüberwachten Trainings des linearen CRFs nützlich ist. Durch diese lassen sich Einfluss der einzelnen Komponenten beispielhaft erläutern und somit die Änderungen der Parameter interpretieren.

Hierfür sei λ_k der Vektor der zu $\Phi_k(s_2, \mathbf{X}, n)$ gehörenden Gewichte sowie λ_{kj} das zu $\Phi_{kj}(s_1, s_2)$ gehörende Gewicht, sodass für λ

$$\lambda = \begin{bmatrix} [\lambda_k]_{k=1}^{K} \\ \left[[\lambda_{kj}]_{k=1}^{K}\right]_{j=1}^{K} \end{bmatrix}$$

und somit für die Potentialfunktion

$$\begin{aligned}\exp\left(\lambda^\top \Phi(s_1, s_2, \mathbf{X}, n)\right) &= \exp\left(\sum_k \lambda_k^\top \Phi_k(s_2, \mathbf{X}, n) + \sum_{k,j} \lambda_{kj} \cdot \Phi(s_1, s_2)\right) \\ &= \exp\left(\sum_k \lambda_k^\top \Phi_k(s_2, \mathbf{X}, n)\right) \cdot \exp\left(\sum_{k,j} \lambda_{kj} \cdot \Phi(s_1, s_2)\right)\end{aligned} \tag{4.57}$$

gilt. Das lineare CRF lässt sich also in diesem Fall in einen Anteil, der den Einfluss der Daten auf die Markov-Kette beschreibt, und einen Anteil, der die Übergänge zwischen den Zuständen beschreibt, zerlegen. Die Funktion

$$f_k(s_2, \mathbf{X}, n) = \exp\left(\lambda_k^\top \Phi_k(s_2, \mathbf{X}, n)\right) \tag{4.58}$$

wird im folgenden *Ladungsfunktion* für die Farbe ζ_k genannt.

Erwünscht ist, dass das Markov-Modell des linearen CRFs optimal ist. Dieses betrifft sowohl die Übergangswahrscheinlichkeiten zwischen den Zuständen als auch die Ladungsfunktionen.

Für die Ladungsfunktionen ist das Ziel zur Beschleunigung der Optimierung des CRFs hin zu einem optimalen Markov-Feld, dass der Erwartungswert einer jeder Ladungsfunktion gleich ist, das heißt

$$E\{f_1(s_2, \mathbf{X}, n)\} = E\{f_2(s_2, \mathbf{X}, n)\} = \cdots = E\{f_K(s_2, \mathbf{X}, n)\}. \tag{4.59}$$

Existiert der Erwartungswert von $\mathbf{x}(n)$, also $E\{\mathbf{x}(n)\}$, so gilt

$$E\{f_k(s_2, \mathbf{X}, n)\} = \exp\left(E\{\lambda_k^\top \mathbf{x}(n)\}\right) = \exp\left(\lambda_k^\top E\{\mathbf{x}(n)\}\right),$$

somit lässt sich die Gleichheit des Erwartungswertes der Ladungsfunktion (4.59) überführen in Bedingungen für den Erwartungswert der Messwerte.

Sei der Einfachheit halber $\mathbf{x}(n) = [x_l(n)]_{l=1}^d$ ein d-dimensionaler Vektor mit

$$E\{x_1(n)\} = E\{x_2(n)\} = \cdots = E\{x_d(n)\},$$

das heißt der Erwartungswert jeder Komponente ist gleich. Dann ist die Gleichheit (4.59) erfüllt, wenn gilt

$$\sum_{l=1}^{d} \lambda_1(l) = \sum_{l=1}^{d} \lambda_2(l) = \cdots = \sum_{l=1}^{d} \lambda_K(l), \tag{4.60}$$

das heißt, wenn die Summen der Gewichte für einen Zustand gleich sind, zum Beispiel $\sum_{l=1}^{d} \lambda_k(l) = 0$. Diese Bedingung kann sowohl in dem Startwert von λ, nach jeder Iteration im Training des CRFs oder als Bedingung in einem Optimierungsschritt [21] verwendet werden [50].

Diese Normierung kann ebenfalls für die Übergangswahrscheinlichkeiten verwendet werden. Damit die Übergangswahrscheinlichkeiten zwischen den Zuständen vergleichbar bleiben, kann es sinnvoll sein, die nicht normierten Ausgangswahrscheinlichkeiten zu kontrollieren. Hierfür werden die Ausgangswahrscheinlichkeiten dadurch kontrolliert, dass für diese die Summe $\sum_{k=1}^{K} \lambda_{jk}$ auf 0 gesetzt wird. Diese Normierung kann derart modelliert werden, dass ein Ereigniszustand konstruiert wird: Mit dieser Methode kann die A-priori-Wahrscheinlichkeit eines Übergangs von einer normalen Farbe in eine Farbe des Ereignisses kontrolliert werden, dadurch wird letztendlich die Sensitivität der Methode kontrolliert.

Diese Bedingung verhindert nicht, dass ein Zustand des CRFs degenerieren kann, das heißt, dass für eine Farbe $\lambda_k(1) = \lambda_k(2) = \cdots = \lambda_k(d) = 0$ gilt, damit die Messung keine Information Einfluss auf die Wahrscheinlichkeit dieses Zustandes besitzt. Dieses lässt sich verhindern, indem die Länge des Gewichtsvektors selbst kontrolliert wird, zum Beispiel die Länge von λ_k nach jedem Optimierungsschritt skaliert mit

$$\lambda_k \leftarrow \lambda_k \cdot \frac{1}{\|\lambda_k\|_2}. \tag{4.61}$$

Diese Normierung verhindert die Degeneration einer Farbe des CRFs und ist somit auch bei unbekanntem Erwartungswert eine sinnvolle Ergänzung des Optimierungsschrittes [49].

Der Trainingsalgorithmus lässt sich folgendermaßen zusammenfassen. Der Gewichtsvektor λ wird zufällig initialisiert. Anschließend werden die Gewichte λ_k und λ_{jk} entsprechend der Normierungen gesetzt, die für das Training gewählt wurden. Mit diesem Gewichtsvektor wird eine optimale Sequenz $\hat{\mathbf{S}}$ geschätzt. An diese Sequenz wird das CRF in einem Trainingsschritt nach der Maximierung der logarithmierten Likelihood angepasst. Daraus folgt ein neuer Gewichtsvektor, der ebenfalls normiert wird.

Anschließend wird mit dem angepassten Gewichtsvektor eine neue optimale Sequenz geschätzt. Dieses Training wird fortgesetzt, bis die Konvergenz erreicht wird.

Das unüberwachte Training eines CRFs ähnelt demnach einem üblichen EM-Algorithmus [21]. Auch wenn die logarithmierte Likelihood eines CRFs eine streng konvexe Funktion ist, ist es die Zielfunktion des EM-artigen Algorithmus nicht.

Das unüberwachte Training wurde insbesondere für die Ereignisdetektion entworfen. Diese ist das zentrale Thema des nächsten Kapitels. Dabei geht es insbesondere um die Modellierung und Interpretation, um ein Ereignis zu detektieren. Die theoretischen Grundlagen und das Training eines Modells wurden in diesem Kapitel vorgestellt. Eine Ergänzung zu einem Training mit Ereigniszustand wird ebenfalls im nächsten Kapitel besprochen. In Kapitel 6 werden anschließend Experimente mit diesen Algorithmen vorgestellt.

4.8 Diskussion zu Maximum-Entropy-Markov-Modellen und bedingten Zufallsfeldern

MEMMs und CRFs sind Modelle, die zur Analyse von Signalen geeignet sind. Ein zeitdiskretes Signal wird dabei segmentiert und somit in eine endliche Anzahl von Fällen unterteilt. Das Training der CRFs erfolgt traditionell anhand von vorgefärbten Graphen, also einer Kombination von Beispielmessungen und gegebenen Zustandssequenzen. Das CRF wird anschließend darauf trainiert, in neuen Sequenzen die bekannten Merkmale zu segmentieren. Dabei wird eine Verteilung über die Zustände bestimmt; die Verteilung der Merkmale muss nicht bekannt sein.

Diese Verfahren bieten mehrere Eigenschaften, die für die Ereignisdetektion nützlich sind. Einmal können so Sensoren direkt kombiniert werden, die zuvor nur sehr umständlich in einem geschlossenen Modell besprochen werden können. Dies folgt aus der Eigenschaft, dass CRFs und MEMMs die Verteilung der Merkmale nicht schätzen. Dadurch werden auch keine falschen Annahmen über die Verteilungen der Merkmale getroffen.

Ferner basieren die CRFs auf dem Prinzip der maximalen Entropie. Selbst wenn die direkten Vergleiche zwischen den hier besprochenen Modellen und herkömmlichen Lösungen deutlich schwieriger sind als bei dem Mischmodellansatz, kann aus dieser Eigenschaft gefolgert werden, dass die Qualität der Modelle ausschließlich noch von der Qualität der Merkmalsextraktion abhängig ist: Die Verteilung der maximalen Entropie ist so konzipiert, dass sie keine zusätzlichen Annahmen macht. Werden also die Informa-

tionen richtig genutzt, ist die Bestimmung der Markov-Kette bestmöglich. Existiert zu einer bestehenden Modellierung eine bessere Segmentierung, kann daraus geschlossen werden, dass die benutzten Annahmen implizit richtig liegen und somit als Merkmale hinzugefügt werden müssen oder dass die Merkmalsextraktion suboptimal ist. Aus diesem Grund erübrigt sich auch für allgemeine Anwendungsbeispiele der Vergleich mit anderen Methoden, nur konkrete Implementierungen zu einem feststehenden Problem sind zu vergleichen.

Die für die Ereignisdetektion notwendige Echtzeitanalyse wurde in diesem Kapitel ebenfalls besprochen. Dies ist eine wichtige und nützliche Erweiterung zu CRFs und MEMMs und verbindet diese in einem geschlossenen Modell. Auch das unüberwachte Training kann ohne die Ereignisdetektion verwendet werden, wenn eine Interpretation der Farbe selbst nicht benötigt wird, zum Beispiel als Zwischenschicht inerhalb eines größeren Modells oder um das CRF zur Merkmalsextraktion oder Segmentierung zu nutzen. In diesen Fällen ist die Interpretation einer Farbe nicht zwingend erforderlich.

CRFs und MEMMs sind nicht ohne Anpassungen für die Ereignisdetektion geeignet. Insbesondere der Umstand, dass die Ereignisse vollständig unbekannt sind, erschwert die einfache Anwendung der bekannten Modelle. Die benötigten Anpassungen werden im nächsten Kapitel besprochen.

5 Bedingte Zufallsfelder zur Ereignisdetektion

Einer der wichtigsten Spezialfälle eines CRFs für die Ereignisdetektion ist das lineare CRF. Diese spezielle Form des Markov-Modells ist hilfreich, wenn ein zeitabhängiges Signal analysiert wird, da ein Zustand einem Zeitpunkt zugeordnet werden kann. Die Ereignisdetektion in der Form, die Gegenstand dieses Kapitels ist, beinhaltet die Analyse eines zeitabhängigen Signals, demnach wird dieser Spezialfall hier betrachtet.

Die Aufgabenstellung dieses Kapitels ist es, Anpassungen für ein CRF zu besprechen, die helfen, das CRF für die Ereignisdetektion zu verwenden. Diese Anpassungen wurden im Rahmen dieser Arbeit entwickelt, sind somit komplett neue Algorithmen. Eine der wesentlichen Anpassungen, die sequentielle Segmentierung, wurde bereits im letzten Kapitel besprochen, da es sich um eine generelle Modifikation handelt, die nicht notwendigerweise auf die Ereignisdetektion beschränkt ist. Dennoch ist sie bei der zur Aufnahme der Daten parallelen Verarbeitung, also bei einer Echtzeitanalyse, nützlich. In diesem Kapitel wird der Umgang mit unbekannten Messungen und die zeitliche Interpretation einer gefärbten Zustandssequenz besprochen.

Im Folgenden sei $\mathbf{X}$ stets ein zeitabhängiges, mehrdimensionales Signal der Länge N, also $\mathbf{X} = \mathbf{x}(1), \mathbf{x}(2), \ldots, \mathbf{x}(N)$. Die begrenzte Länge dieses Signals dient der Vereinfachung, wegen der sequenziellen Segmentierung wird die Anwendung der hiesigen Methoden nicht auf Nicht-Echtzeitanwendungen beschränkt. Zur Analyse des Signals wird eine Markov-Kette derselben Länge modelliert: $\mathbf{S} = s(1), s(2), \ldots, s(N)$. Jedem Messwert des Signals $\mathbf{x}(n)$ wird ein eindeutiger Zustand zugeordnet. Die zeitliche Zuordnung erfolgt durch den Index n.

Das CRF wird derart modelliert, dass eine Potentialfunktion global gültig ist, das heißt, die Funktion Φ in Definition 4.2 vereinfacht sich zu

$$\Phi(s(n-1), s(n), \mathbf{X}, n) = \Phi(s(n-1), s(n), \mathbf{x}(n)). \tag{5.1}$$

Dieses hat den Vorteil, dass die Länge des CRFs N nicht vorher feststeht, das heißt,

wir können vergleichbare Markov-Felder für unterschiedlich lange Signale modellieren, auch wenn diese Einschränkung bei CRFs generell nicht notwendig ist. Theoretisch kann die Potentialfunktion sich in jedem Zeitpunkt ändern; praktisch wird darauf in den meisten Fällen verzichtet, um die Algorithmen handlicher zu halten. Für die Probleme, die im Folgenden besprochen werden, ist dieses Vorgehen unnötig, und daher wird an dieser Stelle diese übliche Einschränkung eingeführt.

Bei der Ereignisdetektion erwartet man für gewöhnlich das Eintreten einer unbekannten, nicht vorhergesehenen Situation zu einem unbekannten Zeitpunkt. Ein Beispiel sei hier die Videoüberwachung. Bei der Überwachung eines Gebietes oder Raumes kann es vorkommen, dass über längere Zeiträume nur Situationen eintreten, die als normal zu bewerten sind, wie tägliche Arbeiten oder sonstiges unauffälliges Verhalten. Die entsprechenden Ereignisse können von Vandalismus, Diebstahl und Einbruch bis hin zu zurückgelassenen Objekten reichen. Aufgrund der Seltenheit und unterschiedlicher Dauer der Ereignisse (Vandalismus und Diebstahl sind spontane Ereignisse, die in nur wenigen Bildern beschrieben sind, über eventuell zurückgelassene Objekte kann jedoch oft nur über größere Zeiträume entschieden werden, ob sie zurückgelassen wurden) ist eine variable Analyse der Zeitsequenzen wünschenswert. Durch die Modellierung einer Zustandssequenz, deren einzelne Zustände $s(n)$ Messwerten $\mathbf{x}(n)$ zugeordnet werden können, kann ferner der Zeitpunkt eines erfolgreich detektierten Ereignisses bestimmt werden, also der Zeitpunkt des Messwertes $\mathbf{x}(n)$ [48, 51].

Ein Ereignis kann durch einen im Vergleich zum Normalfall deutlich unterschiedlichen Messwert festgelegt sein. Allerdings ist es ebenfalls möglich, dass dieses Ereignis erst in einem größeren Zusammenhang deutlich wird. Zum Beispiel kann im Falle einer Videobeobachtung jeder Zustand einer Aktion entsprechen, die eine beobachtete Person ausführt. Ein Ereignis kann sein, dass diese Person für diese Aktion mehr (oder weniger) Zeit benötigt als üblich. Dieses lässt sich zumeist nicht durch die Beobachtung eines einzelnen Bildes der Videosequenz beurteilen, sondern nur in längeren Segmenten.

Im Folgenden werden unterschiedliche Methoden zur Ereignisdetektion mittels eines CRFs diskutiert. Insbesondere wird eine Interpretation längerer Sequenzen betrachtet, als auch speziell zur Ereignisdetektion entworfene CRFs. Die Kombination unterschiedlicher Modelle liefert die Anpassungsfähigkeit dieses Ansatzes.

5.1 Modellierung eines bedingten Markov-Zufallsfeldes mit Ereignisfärbung

Einer der wichtigsten Spezialfälle der Ereignisdetektion ist der, bei dem ein Messwert $\mathbf{x}(n)$ unterschiedlich zu den bekannten Normalfall-Messwerten ist. Dieser unterschiedet sich von dem Fall, dass Abfolgen von Situationen der Datenquellen unterschiedlich sind, jeder Messwert für sich aber einem einzelnen bekannten und normalen Zustand zugeordnet werden kann. Dieser Spezialfall der "herausragenden" Daten lässt sich auch mit Klassifikatoren entdecken, die für einzelne Klassen trainiert sind, zum Beispiel die Single Class SVM [44, 84].

Allerdings ist die Frage, wie sehr der Messwert sich von dem Normalfall unterscheiden muss, bis er einem Ereignis zugeordnet werden kann, abhängig von einem Schwellwert. Oft werden feste Schwellwerte angenommen oder mit Hilfe von Trainingsdatensätzen gelernt. Allerdings ist ein anpassungsfähiger Schwellwert oft nützlicher als ein fester. Zum Beispiel kann ein unwahrscheinlicher Übergang, kombiniert mit einem Messwert, bereits ein Ereignis sein, auch wenn sich der Messwert $\mathbf{x}(n)$ ohne diese Information aus seinen Nachbarn nicht deutlich genug unterscheidet. Darum ist die Verwendung eines Markov-Modells zur Ereignisdetektion, das diese Übergänge mit einbezieht, nützlich. Mit Hilfe der Markov-Modelle kann ein größerer Überblick über die möglichen Ereignisse gewonnen werden: Sie bieten die Möglichkeit, die Ereignisse in einem generelleren Umfeld zu betrachten. Dennoch kann für diesen Fall der spontanen Ereignisse zunächst die Eigenschaft der Markov-Kette ignoriert und nur die einzelne Potentialfunktion betrachtet werden. Das bedeutet, dass in diesem Abschnitt die Wahrscheinlichkeit für einen Zustand bei gegebenen Messwert $\mathbf{x}(n)$ betrachtet werden kann, also $p(s(n)|\mathbf{x}(n))$, ohne Betrachtung der Nachbarschaftsbeziehungen, das heißt

$$P(s(n) = \zeta_k|\mathbf{x}(n)) = \frac{1}{Z(\mathbf{x}(n))} \exp\left(\lambda_k^\top \phi_k\left(\mathbf{x}(n)\right)\right), \tag{5.2}$$

wobei λ_k der Vektor der Gewichte für den Zustand ζ_k ist und $Z(\mathbf{x}(n))$ die Normalisierungskonstante, $Z(\mathbf{x}(n)) = \sum_{\zeta_l} \exp\left(\lambda_l^\top \phi_l\left(\mathbf{x}(n)\right)\right)$. Die Eingliederung in ein Zufallsfeld ist durch die mögliche Zerlegung von CRFs, die im letzten Kapitel besprochen wurde, gegeben.

Das CRF definiert die Wahrscheinlichkeit für jeden möglichen Zustand zu jedem Zeitpunkt, ähnlich wie ein HMM, das jedem der versteckten Zustände eine Wahrscheinlichkeit zuordnet. Es gilt in beiden Fällen die Randbedingung, dass zu jedem Zeitpunkt

die Summe der Wahrscheinlichkeiten aller Zustände eins ist. Während bei HMMs die Verteilung der Messwerte modelliert wird und dadurch implizit (oft falsche) Annahmen über die Messwerte getätigt werden, werden die Messwerte bei CRFs für jeden Zustand gewichtet aufsummiert (vgl. Definition 4.2).

Um ein CRF zur Ereignisdetektion zu verwenden, müssen zwei Punkte behandelt werden, das ist das Design des Modells und die Interpretation der Zustände. Beides beeinflusst sich gegenseitig: Bestimmte Designs des CRFs lassen unterschiedliche Interpretationen zu. Hierbei ist auch zu bedenken, welche Ereignisse detektiert werden sollen. Im Folgenden werden Modelle und Interpretationsmöglichkeiten diskutiert.

Ein spontanes Ereignis kann im Wesentlichen durch zwei Methoden mit Hilfe eines CRFs, das nur auf den Normalfall trainiert wird, detektiert werden. Zunächst bedeutet ein Ereignis, dass der Messwert $\mathbf{x}(n)$ sich von den vorherigen Messwerten deutlich unterscheidet. Er soll folglich keinem der bekannten, für den Normalfall definierten Zustände zugeordnet werden: Dadurch ist die Wahrscheinlichkeit für jeden Zustand gleich, denn das neue Merkmal liefert keine Information zu einem der Zustände.

Ein Maß für diese Gleichheit ist die Entropie zu dem gegebenen Zeitpunkt

$$h(n) = -\sum_{k=1}^{K} P(s(n) = \zeta_k|\mathbf{x}(n)) \cdot \log\left(P(s(n) = \zeta_k|\mathbf{x}(n))\right). \tag{5.3}$$

Die einfachste Interpretation ist mit Hilfe eines Schwellwerts. Überschreitet diese Entropie einen Schwellwert θ, ist ein Ereignis erkannt. Dieses Verfahren ist durchaus naheliegend, allerdings ist es abhängig von der Skalierung der Gewichtsvektoren. Angenommen, wir klassifizieren nach der Verteilung definiert in (5.2) und Gewichtsvektoren λ_k für $k = 1, 2, \ldots, K$. Dann erhalten wir dieselbe Entscheidung mit Gewichtsvektoren $const \cdot \lambda_k$, allerdings verändert sich die Entropie mit dieser Skalierung, wenn keine exakte Gleichverteilung erreicht ist, wie aus der Definition 4.2 (bedingte Zufallsfelder) abgeleitet werden kann. Ebenfalls ist die Entropie abhängig von der Skalierung von $\mathbf{x}(n)$, was eine Schätzung eines Schwellwerts $\hat{\theta}$ erschwert.

Im folgenden Beispiel wird die Abhängigkeit von der Skalierung des Gewichtsvektors beziehungsweise des Merkmals gezeigt. Sei in diesem Beispiel $\mathbf{x} = [1, 0, 1]^\top$ und $\lambda = [1, -1, 0, -2, 1, 1]$ sowie $\phi(\mathbf{x}, \zeta_1) = [\mathbf{x}, 0, 0, 0]^\top$ und $\phi(\mathbf{x}, \zeta_2) = [0, 0, 0, \mathbf{x}]^\top$. Auf eine Abhängigkeit zu Nachbarzustände wird zugunsten der Übersichtlichkeit verzichtet. Untersucht wird eine Skalierung des Parametervektors $c \cdot \lambda$, da eine Skalierung des

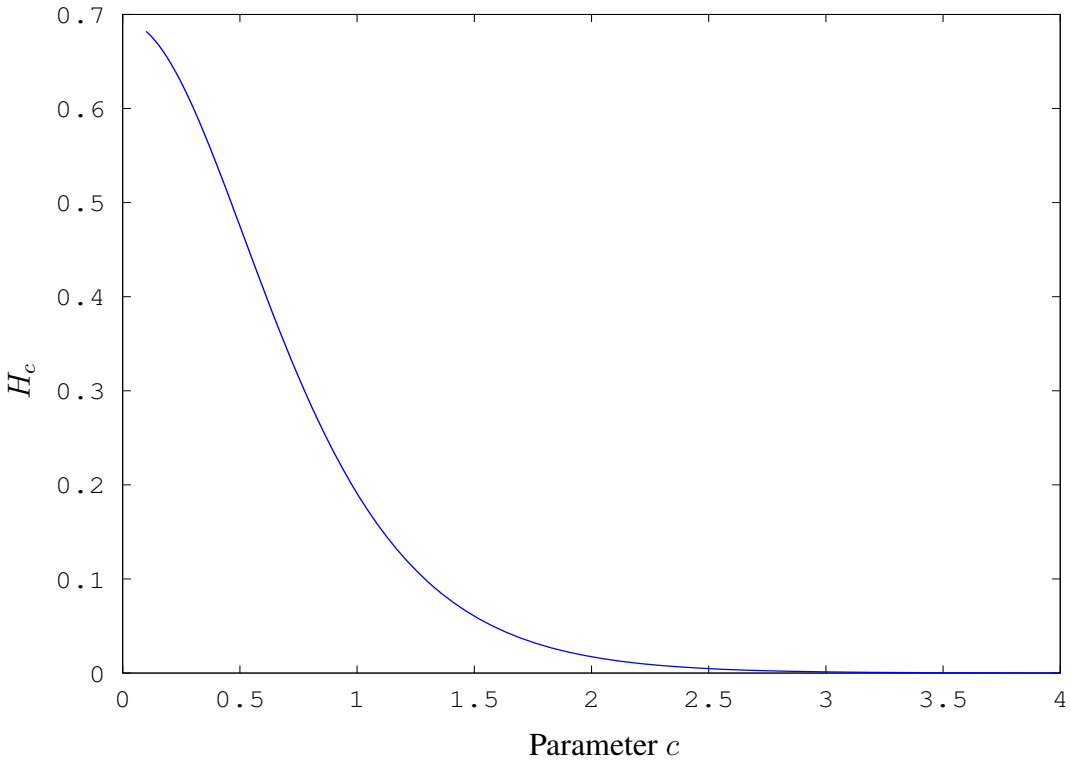


Abbildung 5.1: Beispiel: die Entropie ist abhängig von der Skalierung.

Vektors $\mathbf{x}$ hierzu äquivalent ist. Demnach ist entsprechend der Definition des CRFs die von c abhängige Wahrscheinlichkeit der beiden Zustände gegeben durch

$$P_c(s = \zeta_1|\mathbf{x}) = \frac{1}{\exp(c) + \exp(-2c)} \exp(c),$$
$$P_c(s = \zeta_2|\mathbf{x}) = \frac{1}{\exp(c) + \exp(-2c)} \exp(-2c).$$

Die von diesen Wahrscheinlichkeiten abhängige Entropie ist

$$H_c(\mathbf{x}) = -\left(P_c(s = \zeta_1|\mathbf{x}) \cdot \log(P_c(s = \zeta_1|\mathbf{x})) + (P_c(s = \zeta_2|\mathbf{x}) \cdot \log(P_c(s = \zeta_2|\mathbf{x})))\right). \tag{5.4}$$

Diese Entropie für diesen speziellen Messwert ist in Abhängigkeit vom Parameter c in Abbildung 5.1 dargestellt. Da diese nicht konstant ist, muss jeder Schwellwert, der hier verwendet wird, ebenfalls von dem Parameter c abhängig sein; dieses kann sich in der Praxis als schwierig erweisen.

Aus diesem Grund wird in dieser Arbeit ein anderer Ansatz verfolgt. Dabei wird das Markov-Modell selbst modifiziert. Neben den K Farben, die den Normalfall beschreiben, wird eine zusätzliche Farbe ζ_0 eingefügt. Diese Farbe wird angenommen, wenn der Messwert $\mathbf{x}(n)$ sehr unterschiedlich zu den gelernten Messwerten ist. Diese Farbe ist somit eine Ereignisfarbe.

5.1.1 Informationsextraktion

Die Funktion $\Phi(s(n-1), s(n), \mathbf{x}(n))$ sei im Folgenden definiert als [48, 51]

$$\Phi(s(n-1), s(n), \mathbf{x}(n)) = \begin{bmatrix} [[\![s(n) = \zeta_k]\!] \cdot \mathbf{x}(n)]_{k=1}^{K} \\ \left[[[\![s(n-1) = \zeta_j]\!] \cdot [\![s(n) = \zeta_k]\!]]_{k=1}^{K} \right]_{j=1}^{K} \\ [\![s(n) = \zeta_0]\!] \cdot \mathbf{x}(n) \end{bmatrix}, \tag{5.5}$$

das heißt, dass sich die Funktion in drei Teilen interpretieren lässt: erstens der Zusammenhang zwischen Messwerten und dem Normalfall, zweitens der Übergang zwischen den Zuständen und drittens der Zusammenhang der Messwerte mit dem Ereignis. Jeder dieser Teile entspricht einer Zeile in (5.5). Zunächst wird der Zusammenhang zwischen den Zuständen und den Merkmalen besprochen.

CRFs werden oft mit binären Datenvektoren erklärt [28, 41]. Der Grund wird deutlich, wenn man die Potentialfunktion $f(s(n-1), s(n), \mathbf{x}(n))$ betrachtet:

$$f(s(n-1), s(n), \mathbf{x}(n)) = \exp\left(\lambda^{\top} \Phi(s(n-1), s(n), \mathbf{x}(n))\right).$$

Die Skalierung von jeder Komponente von $\mathbf{x}(n)$ hat Einfluss auf die Potentialfunktion und damit letztendlich auf die Likelihood. Mit der Berücksichtigung dieses Umstands, zum Beispiel derart, dass jede Komponente von $\mathbf{x}(n)$ nur entweder 0 oder 1 ist, ist die Interpretation von λ wesentlich eindeutiger: je größer der Betrag einer Komponente von λ, desto wichtiger ist die entsprechende Komponente von $\mathbf{x}(n)$ für die Bestimmung der Wahrscheinlichkeit für eine Farbe. Ist $\mathbf{x}(n)$ hingegen schlecht skaliert, fehlt diese Interpretation.

Bei der Ereignisdetektion tritt ein weiteres Problem auf. Ein häufiger Fall ist es, dass eine Komponente von $\mathbf{x}(n)$, also $x_i(n)$ bei gegebenen Normalfall zwischen einem Minimum und einem Maximum liegt, also $x_{i,min} < x_i(n) < x_{i,max}$, und ein Über- beziehungsweise Unterschreiten dieser Grenzen, nicht ausschließlich, ein Ereignis impliziert. Die Skalierung sowie das Vorzeichen dieses Intervalls hat offensichtlich Einfluss auf die Entscheidung.

Als Lösung hierfür können Merkmale zunächst transformiert werden, um testbare Informationen zu erhalten, wie sie zur Bestimmung einer Verteilung maximaler Entropie, siehe Abschnitt 4.2, verwendet werden. Hierbei sei angemerkt, dass diese Transformation auch für übliche CRFs sinnvoll sind, nicht nur für CRFs zur Ereignisdetektion; allenfalls die Form der Transformation unterscheidet sich. Praktische Beispiele werden in Kapitel

6 behandelt. Diese Transformation dient der Informationsextraktion und kann demnach im Allgemeinen auch nicht erschöpfend behandelt werden, da diese vom konkreten Problem abhängig sein kann. Hier wird eine einfache und effektive Methode beschrieben, die zum Beispiel in [51] verwendet wurde.

Der Einfachheit halber sei die Messung $y(n) \in \mathbb{R}$ ein eindimensionales Signal. Mehrdimensionale Signale können dadurch behandelt werden, dass auf jede Komponente einzeln dieselbe Transformation angewendet wird. Sei ferner $y_{min} < y(n) < y_{max}$ für jede Messung $y(n)$, die zum Normalfall gehört. Diese Grenzen lassen sich für praktische Probleme mit Hilfe des Minimums respektive Maximums der Trainingsmenge schätzen: es wird angenommen, dass die Trainingsmenge den Normalfall beschreibt. Dazu gehören auch die Schranken. Sei $y_l = y_{min} + (y_{max} - y_{min}) \cdot \frac{l-2}{L-2}$ für $L \in N$ und $0 < \vartheta$.

Dann kann das Signal in ein L-dimensionales Signal transformiert werden durch

$$x_l(n) = \begin{cases} [\![y(n) < y_{min}]\!] \cdot \exp(y_{min} - y(n)) - \vartheta & \text{falls } l = 1 \\ [\![y_l \leq y(n)]\!] \cdot [\![y(n) < y_{l+1}]\!] & \text{falls } 1 < l < L \\ [\![y_{max} \leq y(n)]\!] \cdot \exp(y(n) - y_{max}) - \vartheta & \text{sonst} \end{cases} \tag{5.6}$$

und $\mathbf{x}(n) = [x_l(n)]_{l=1}^{L}$.

Diese Transformation hat für die Ereignisdetektion folgende Vorteile. Gehört eine Messung $y(n)$ zum Normalfall, so liegt diese zwischen den beiden Grenzen y_{min} und y_{max}. Dadurch ist genau eine Komponente dieses Vektors positiv. Die erste und letzte Komponente, die das Signal außerhalb des normalen Bereiches beschreiben, sind negativ. Dieses trifft für die Trainingsmenge immer zu, da die Grenzen entsprechend definiert wurden. Weicht das Signal von diesem Bereich ab, so wächst eine der Komponenten, die für gewöhnlich negativ sind, exponentiell. Dadurch wird die Eigenschaft eines CRFs, von der Skalierung beeinflussbar zu sein, ausgenutzt. Die Konstante ϑ kontrolliert den Einfluss von einzelnen Messungen, die die Grenzwerte über- beziehungsweise unterschreiten. Generell ist $\vartheta = 0.1$ eine praktikable Wahl, die sich für viele Probleme anwenden lässt.

Diese Form der Quantisierung ist eine praktische Methode, die Stetigkeit von Signalen für CRFs zu übersetzen [51]. Sie ist dadurch auch für Fälle anwendbar, bei denen ein standardmäßiges CRF auf stetige Daten angewendet werden soll.

Bei einer Beobachtung ist immer nur eine der Komponenten positiv. Der Einfachheit halber wird darum im Folgenden von dem Eintreten einer Beobachtung gesprochen: eine Beobachtung tritt ein, wenn genau eine Komponente positiv ist, also bei einer Instanz

des Messwerts. Dieses gilt auch für die Beobachtung, die außerhalb des Bereichs liegt, der durch y_{min} und y_{max} definiert ist. Die Beobachtungen treten mit unterschiedlichen Häufigkeiten ein. Durch das Training wird der Zusammenhang zwischen dem Eintreten der Beobachtungen und der Färbung des Graphen festgelegt.

Der zweite Abschnitt der Potentialfunktion, der Übergang zwischen den Zuständen, unterscheidet sich nicht von anderen linearen CRFs. Hier sind die Übergänge zu den bekannten Zuständen beschrieben. Der dritte Abschnitt entspricht dem ersten, mit dem Unterschied, dass er den Zusammenhang zwischen den Merkmalen und dem Ereignis beschreibt. Hier sind die Randbereiche besonders interessant: Seltene Merkmale, wie die erste und letzte Komponente, sollen nach dem Training positive Gewichte im Ereigniszustand haben. Das Training wurde dementsprechend modifiziert. Tritt nun ein solches Merkmal ein, so steigt die Wahrscheinlichkeit für ein Ereignis. Dieses ist das Prinzip des CRFs mit Ereignis-Färbung.

5.1.2 Training eines bedingten Markov-Zufallsfeldes mit Ereignisfärbung

Die bisherigen Trainingsalgorithmen für CRFs diskutierten nur den Fall, dass für alle Farben Beispiele in der Trainingsmenge vorhanden sind. Das unüberwachte Training teilt die Menge der Trainingsdaten in einzelne Klassen auf, dadurch sind auch hier für alle Farben Beispiele in der Trainingsmenge enthalten.

Bei der Ereignisdetektion tritt der Fall ein, dass Beispiele für das Ereignis nicht in der Trainingsmenge enthalten sind. Wird das CRF demnach mit Hilfe einer speziellen Färbung für das Ereignis definiert, so sind Modifikationen an dem Training unerlässlich [48, 51]. Informationen über die Struktur des Ereignisses werden aus dem Normalfall geschätzt. Für ein solches Training werden im Folgenden zwei Möglichkeiten erläutert. Dabei handelt es sich um zwei alternative Methoden, das Ereignis zu trainieren. Im einfacheren Fall ist eine bedingte Optimierung ausreichend, allerdings müssen hierfür Annahmen an die Daten respektive die Merkmalsextraktion gestellt werden. Der andere Fall ist für komplexere Daten geeignet und bietet deutlich mehr Möglichkeiten, das Training anzupassen. In beiden Fällen wird die Funktion in (5.5) als definierende Funktion $\Phi(s(n-1), s(n), \mathbf{x}(n))$ verwendet. Eine Erweiterung auf andere Funktionen ist möglich, bedingt jedoch Anpassungen der hier vorgestellten Methoden.

Im einfachen Fall lassen sich die Parameter des Ereigniszustands mittels einer bedingten Optimierung [21] bestimmen, wie im Folgenden gezeigt wird. Dabei wird eine äquivalente Zerlegung der Funktion wie in dem blinden Training in Abschnitt 4.7.2 verwendet [48]. Es sei

$$\begin{aligned}\Phi_k(s_2, \mathbf{x}(n)) &= [\![s_2 = \zeta_k]\!] \cdot \mathbf{x}(n) \\ \Phi_{kj}(s_1, s_2) &= [\![s_1 = \zeta_j]\!] \cdot [\![s_2 = \zeta_k]\!]\end{aligned}$$

für $k, j = 0, 1, 2, \ldots, K$. Der Unterschied zu der Zerlegung im Fall des blinden Trainings ist, dass hier auch die Farbe ζ_0 betrachtet wird: diese repräsentiert das Ereignis, ist also nicht in der Trainingsmenge enthalten. Dementsprechend sei auch λ_k und λ_{kj} eine Zerlegung von λ, sodass gilt

$$\lambda^\top \Phi(s_1, s_2, \mathbf{x}(n)) = \sum_{k=0}^{K} \lambda_k^\top \Phi_k(s_2, \mathbf{x}(n)) + \sum_{k,j=0}^{K} \lambda_{kj} \cdot \Phi_{kj}(s_1, s_2). \tag{5.7}$$

Das Training des CRFs mit einer Ereignisfärbung basiert auf dem Gradientenverfahren, das bedeutet, λ wird angepasst gemäß der Regel

$$\lambda \leftarrow \lambda + c \cdot \left(\sum_{n=1}^{N} \Phi(s(n-1), s(n), \mathbf{x}(n)) - E_{\Phi(\hat{\mathbf{S}}, \mathbf{X})} \left\{ \Phi(\hat{\mathbf{S}}, \mathbf{X}) | \mathbf{X} \right\} - \frac{1}{2\sigma} \lambda \right),$$

wobei c die Schrittlänge ist. In diesem wird der Erwartungswert der Funktion Φ an eine Testmenge, für die die Sequenz bekannt ist, angepasst. Tritt eine Beobachtung $x_l(n)$ besonders oft auf wenn gilt $s(n) = \zeta_k$, so wird das entsprechende Gewicht $\lambda_k(l)$ ebenfalls positiv. Umgekehrt gilt demnach auch, dass ein positives Gewicht einer hohen Likelihood für die Farbe ζ_k entspricht, wenn diese Beobachtung eintritt. Tritt andernfalls diese Beobachtung ausschließlich oder vornehmlich bei anderen Färbungen ein, so wird im Training das entsprechende Gewicht negativ. Diese Eigenschaft wird für die Bestimmung der Gewichte für das in der Trainingsmenge nicht repräsentierte Ereignis genutzt. Sie lässt sich aus dem Training von CRFs ableiten [28].

Das Ereignis ist definiert als die Abwesenheit des Normalfalls. Wenn der Normalfall eintritt, existieren oft Beobachtungen, die besonders oft eintreten. Andererseits sind besonders seltene Beobachtungen Indizien für das Ereignis. Die Likelihood für das Ereignis wird im Training anhand der Auftrittshäufigkeit der Beobachtungen angepasst. Gewichte für besonders häufig eintretende Beobachtungen werden negativ, da diese auf

andere Färbungen, die den Normalfall beschreiben, hinweisen. Dieses geht bereits in die herkömmlichen Trainingsverfahren ein, um die Gewichte auf die Farben zu verteilen [28]. Zur Anpassung der Gewichte für besonders seltene Beobachtungen wird die Bedingung gesetzt, dass gilt

$$\sum_{l=1}^{d} \lambda_0(l) = 0, \tag{5.8}$$

das heißt, dass die Summe der Gewichte, die das Ereignis beschreiben sollen, ist 0. Da das Ereignis im Training nie beobachtet wird, werden in der Trainingsmenge häufig beobachtete Komponenten des Gewichtsvektors negativ; durch diese Nebenbedingung werden Gewichte für seltene Komponenten positiv. Diese Nebenbedingung verteilt demnach Informationen über die Häufigkeit von Beobachtungen. Man beachte, dass dieses Vorgehen dem des blinden Trainings in Abschnitt 4.7.2 entspricht.

Das Training mit der Restriktion (5.8) ist eine einfache und effektive Methode, den in der Trainingsmenge nicht enthaltenen Ereigniszustand mit zu trainieren. Die Einschränkungen dienen dazu, die Skalierungen zu kontrollieren und so die Gewichte für das Ereignis bestimmbar zu machen. Ebenfalls sind die Gewichte für die Übergänge zu normalisieren wie im Kapitel 4.7.2 erläutert, da dadurch ein Übergang von einer Färbung des Normalfalls in eine andere, der im Training selten vorkommt, unwahrscheinlicher ist als in das Ereignis.

Diese einfache Methode ist allerdings nicht sehr anpassungsfähig. Sind weitere Informationen bekannt, zum Beispiel, dass eine Beobachtung an sich besonders oft eintritt, selbst wenn diese zum Ereignis gehören kann, so ist ihre Information geringer zu gewichten als andere. Für aufwendigere Situationen empfiehlt es sich daher, eine andere Methode des Trainings zu verwenden. Eine derartige Methode wird im Folgenden vorgestellt. Der Einfachheit halber wird allerdings zunächst nicht davon ausgegangen, dass ein derart komplexer Fall vorliegt; anschließend werden Modifikationen erläutert, in welcher Form einfache Änderungen an dem folgenden Verfahren möglich sind, um komplexere Datenstrukturen und Informationen über diese abzubilden.

Ausgang für dieses Training ist ebenfalls das Gradientenverfahren. Hierbei wird eine ähnliche Modifikation benutzt wie bei anderen Trainingsverfahren für CRFs, das ist die Verwendung eines Strafterms [51]. Wie bereits bei dem Training mittels Restriktion wird die Information der Farben, die zum Normalfall gehören, genutzt, um die Eigenschaften des Ereignisses zu bestimmen, indem man ausnutzt, dass das Ereignis die Abwesenheit des Normalfalls ist.

Hierbei wird die Aktualisierungsregel

$$\lambda \leftarrow \lambda + c \cdot \left(\sum_{n=1}^{N} \Phi(s(n-1), s(n), \mathbf{x}(n)) - E_{\Phi(\hat{\mathbf{S}},\mathbf{X})} \left\{ \Phi(\hat{\mathbf{S}}, \mathbf{X}) | \mathbf{X} \right\} - \frac{1}{2\sigma} \mathbf{D} \cdot \lambda \right) \tag{5.9}$$

verwendet, wobei $\mathbf{D}$ eine quadratische Matrix ist und *Strafmatrix* [51] genannt wird. Man beachte, dass dieses eine Verallgemeinerung des üblichen Trainings ist: gilt $\mathbf{D} = \mathbf{I}$, wobei $\mathbf{I}$ die Einheitsmatrix ist, ist dieses der bereits bekannte Strafterm [28]. Die zu maximierende Funktion ist demnach

$$F(\lambda) = \log\left(p(\mathbf{S}|\mathbf{X}; \lambda)\right) - \frac{1}{\sigma} \cdot \lambda^{\top} \mathbf{D} \lambda,$$

es handelt sich somit um einen quadratischen Strafterm, da λ quadratisch in diesen eingeht. Besonders seltene Merkmale sollen demnach mit betragsmäßig kleinen Gewichten in λ korrespondieren.

Zum Training der Ereignisfärbung wird eine Strafmatrix verwendet, die Informationen über die Gewichte anderer Färbungen sowie die Auftrittshäufigkeiten von Beobachtungen selbst nutzt. Zugleich wird die Information über alle Gewichte in die Ereignisfarbe integriert.

Die Parameter für den Normalfall und für die Übergänge zwischen den Färbungen des Normalfalls, das sind die ersten $d \cdot K + K^2$ Elemente des Vektors λ, werden trainiert wie in den anderen Verfahren. Das bedeutet, dass die Strafmatrix $\mathbf{D}$ eine obere linke Submatrix der Größe $d \cdot K + K^2 \times d \cdot K + K^2$ hat, die einer Einheitsmatrix entspricht.

Aufgrund der Struktur der Funktion Φ in (5.5) sind die Gewichte für ζ_k, also λ_k, im Gewichtsvektor λ an den Positionen $d(k-1)+1$ bis dk für $k = 1, 2, \ldots, K$. Somit sind die Gewichte, die den Zusammenhang zwischen dem Normalfall und dem Merkmal $x_l(n)$ beschreiben, an den Positionen $d(k-1)+l$ für $k = 1, 2, \ldots, K$ und $l = 1, 2, \ldots, d$. Sind diese Werte im Gewichtsvektor λ allesamt positiv, bedeutet das, dass die Auftrittswahrscheinlichkeit dieses Merkmals bei gegebenen Normalfall sehr hoch ist; dadurch soll das entsprechende Gewicht für das Ereignis sehr niedrig sein; dieses Gewicht soll also in der Evolution des Trainings negativ werden. Hierfür sei der Eintrag der Matrix $\mathbf{D}$ an der Position (i, j), das heißt $D(i, j)$, gesetzt auf $\frac{1}{N}$ wenn gilt $i > dK + K^2$ und $i = j + dK + K^2 - d(k-1)$ für $k = 1, 2, \ldots, K$. Es wird demnach wie im einfacheren Verfahren die Information über die Häufigkeit im Normalfall auf das

Ereignis verteilt, nur geschieht dieses in diesem Verfahren durch den Strafterm anstelle mittels einer Nebenbedingung.

Diese Parameter, die den Zusammenhang zwischen häufig auftretenden Merkmalen und dem Ereignis definieren, stehen sehr selten auftretenden Parametern gegenüber. Praktisch ist daher ein Ausgleich der Parameter: Ist die Summe der Beträge der häufig auftretenden Parameter groß, so sollte auch die Summe der Beträge der selten auftretenden Parametern groß sein. Zugleich muss darauf geachtet werden, dass die Submatrix von $\mathbf{D}$, die nur mit Gewichten multipliziert werden, die zum Ereignis gehören, diagonaldominant ist, damit das Verfahren konvergiert, da dann der quadratische Strafterm das Gewicht limitiert. Das heißt, dass die Einträge für $i, j > dK + K^2$ entsprechend gesetzt werden müssen. Hier werden die Einträge auf $D(i,j) = \frac{1}{2d+1}$ für $i \neq j$ und $D(i,j) = \frac{d}{2d+1}$ für $i = j$ gesetzt. Dieses sorgt für eine Verteilung der Gewichtung innerhalb des Ereigniszustands bei gleichzeitiger Konvergenz des Lernalgorithmus: Steigt ein Gewicht in einem Iterationsschritt, so sinken andere Gewichte entsprechend im nachfolgenden Schritt.

Die quadratische Strafmatrix $\mathbf{D} = [D(i,j)]_{i,j=1}^{dK+K^2+d}$ ist hiermit

$$D(i,j) = \begin{cases} 1 & \text{falls } i = j,\ i,j \leq d \cdot K + K^2, \\ \frac{1}{N} & \text{falls } i > d \cdot K + K^2, \\ & \quad i = j + d \cdot K + K^2 - d \cdot (k-1),\ k = 1, 2, \ldots, K \\ \frac{d}{2 \cdot d+1} & \text{für } i,j > d \cdot K + K^2,\ i = j \\ \frac{1}{2 \cdot d+1} & \text{für } i,j > d \cdot K + K^2,\ i \neq j \\ 0 & \text{sonst.} \end{cases} \tag{5.10}$$

Diese Strafmatrix ist, wie oben erläutert, in einigen Aspekten empirisch gewählt und nicht allgemeingültig. Der Vorteil der Verwendung einer Strafmatrix gegenüber dem Trainingsverfahren mittels bedingter Optimierung ist, dass in dem Fall, dass mehr Information über die Signifikanz einzelner Merkmale für das Ereignis vorliegen, die Matrix angepasst werden kann. Wenn zum Beispiel ein Merkmal auch dann sehr häufig positiv ist, wenn das Ereignis vorliegt, und diese Information bereits vor dem Training des Modells bekannt ist, so kann für dieses Merkmal anstelle von $\frac{1}{N}$ ein kleinerer Wert gewählt werden. Diese Matrix ist somit eine Möglichkeit, A-priori-Wissen in das Modell mit einfließen zu lassen.

Derartige Anpassungen sind allerdings erst durch eingehende Analyse des Problems möglich. Für viele Fälle ist das Verfahren mittels bedingter Optimierung ausreichend und aufgrund der einfacheren Verwendung vorzuziehen. In Experimenten hat sich die

prinzipielle Anwendbarkeit beider Trainingsalgorithmen gezeigt. Diese Experimente sind in Kapitel 6 enthalten.

5.2 Statistische Interpretation und Sequenzanalyse zur Ereignisdetektion

Ein CRF wird durch die Funktion Φ, dem üblicherweise im Training erhaltenen Gewichtsvektor λ, den Daten $\mathbf{X}$ und dem Zufallsfeld $\mathbf{S}$ definiert. Eine zur Ereignisdetektion nützliche Funktion Φ wurde bereits erläutert, ebenfalls zwei Verfahren, das CRF für die Ereignisdetektion mit einer speziellen Farbe für das Ereignis zu trainieren. Des Weiteren lässt sich auch ein standardmäßiges CRF zur Ereignisdetektion nutzen, wenn die Sequenz selbst analysiert wird. Beide Möglichkeiten werden im Folgenden erläutert.

Ist das CRF mit einer speziellen Ereignisfärbung versehen, so existiert für jeden Zeitpunkt n eine Wahrscheinlichkeit dafür, dass ein Zustand die Ereignisfärbung annimmt, diese Wahrscheinlichkeit ist $P(s(n) = \zeta_0|\mathbf{X})$. Die einfachste Methode, diese Wahrscheinlichkeit zur Ereignisdetektion zu nutzen, ist die Nutzung eines Schwellwertes. Dafür wird diese Wahrscheinlichkeit mit einem vorher festgelegten Schwellwert verglichen. Ist die Wahrscheinlichkeit höher als der Schwellwert, so wird angenommen, dass ein Ereignis detektiert wurde. Dieser Schwellwert lässt sich mittels der Trainingsdaten bestimmen, zum Beispiel, dass nicht mehr als eine vorher festgelegte Anzahl an Trainingsdaten fälschlicherweise als Ereignis klassifiziert werden sollen. Der Aufwand der Ereignisdetektion mit dem trainierten Modell ist in diesem Fall vergleichbar mit der mittels einer Verteilungsschätzung.

Eine weitere Möglichkeit, mittels eines trainierten CRFs Ereignisse zu detektieren, ist eine Sequenzanalyse. Mit einem CRF kann eine Sequenz $\mathbf{S}$ geschätzt werden, zum Beispiel derart, dass zu jedem Zeitpunkt die Färbung mit der höchsten Likelihood genommen wird. Diese Sequenz kann an sich zur Detektion von Ereignissen genutzt werden. Dafür werden Statistiken auf die Sequenz selbst angewendet. Hier zeigt sich auch, dass das blinde Training eine Verwendung finden kann, selbst wenn eine direkte Interpretation der Farben versagt bleibt: die statistischen Eigenschaften sind auch bei dem unüberwachten Training vorhanden.

Das Prinzip der Sequenzanalyse zur Ereignisdetektion, das im Folgenden diskutiert wird, besteht darin, Zeiten zu messen. Die Annahme ist, dass Aktionen eine mehr oder weniger feste Zeit benötigen, wenn sie ausgeführt werden. In der Sequenz lassen sich diese Zeiten messen, indem man die Länge der Segmente gleicher Farbe misst. Ist diese

signifikant kürzer oder länger als für Segmente von Trainingssequenzen, wurde ein Ereignis detektiert. Dieses Verfahren ist insbesondere im Fall des Überwachungsproblems nützlich, wo angenommen werden muss, dass die beobachtete Person versucht, ihre Tätigkeiten zu verschleiern. Dadurch sind Verfahren, die auf der Detektion suspekter Gegenstände beruhen, oft nicht hilfreich. Vergleichbar schwierig ist die Detektion verdächtiger Bewegungen, die ebenfalls teilweise verdeckt sein können. Aktionen, die zum Ereignis gehören, können eventuell durch die benötigte Zeit gemessen werden.

Da die Motivation bei Verhalten, das nicht dem Normalfall entspricht, eine andere als bei dem normalen Verhalten ist, werden zwangsläufig einige Aktionen mehr oder weniger Zeit beanspruchen. Ein CRF kann folglich darauf trainiert werden, die Aktionen, die zu dem Normalfall gehören, zu klassifizieren. Hierfür kann auch das unüberwachte Training genommen werden, das in Abschnitt 4.7.2 beschrieben wird, da die explizite Aktion, in der das suspekte Verhalten detektiert wurde, für das Problem der Ereignisdetektion nicht relevant ist. Eine Farbe entspricht damit einer detektierbaren Aktion.

Nach dieser Segmentierung wird die Verteilung der Länge der Segmente gleicher Farbe in der Trainingsmenge geschätzt. Ist bei der Klassifikation neuer Aufnahmen die Länge des Segments gleicher Farbe unwahrscheinlich, wird das als Ereignis gemessen. Dieses Verfahren lässt sich auch mit dem CRF anwenden, das eine Ereignisfarbe besitzt. In diesem Fall ist ein Ereignis detektiert, wenn entweder die Wahrscheinlichkeit für die Länge einer Aktion geringer als ein zuvor festgelegter Schwellwert ist oder die Wahrscheinlichkeit für die Ereignis-Färbung hoch ist. Dadurch existiert eine geschlossene Methode zur Ereignisdetektion, die viele unterschiedliche Ereignisse detektieren kann.

5.2.1 Interpretation mit unvollständigen Daten

Eine besondere Eigenschaft der CRFs als datengetriebene Modelle ist es, dass sie auch dann noch eine Klassifikation ermöglichen, wenn eine Messung unvollständig ist. Das ist insbesondere bei Sensornetzwerken eine wichtige Eigenschaft. Bei solchen Netzwerken kann es ein Problem sein, dass einer der Sensoren ausfällt [48]. In diesem Fall kann die Klassifikation ausschließlich mittels der aktiven Sensoren durchgeführt werden.

Angenommen, es existieren L Sensoren. Der Merkmalsvektor von Sensor l zum Zeitpunkt n sei $\mathbf{x}^{(l)}(n)$. Der gesamte Merkmalsvektor ist demnach $\left[\mathbf{x}^{(l)}(n)\right]_{l=1}^{L}$. Ferner sei der Merkmalsvektor des Sensors, der ausgefallen ist, $\mathbf{x}^{(m)}(n)$ mit $1 \leq m \leq L$. Auf diese Daten sei ein CRF trainiert. Der Ausfall des Sensors sei dadurch bemerkbar, dass $\mathbf{x}^{(m)}(n)$ nicht definiert ist, das heißt, entsprechende Daten seien nicht vorliegend.

Für die Auswertung soll die Messung des ausgefallenen Sensors nicht berücksichtigt

werden. Durch die Definition des CRFs zeigt sich, dass nur die Elemente des Vektors λ in die Auswertung eingehen, deren korrespondierende Werte von $\Phi(s(n-1), s(n), \mathbf{x}(n))$ verschieden von 0 sind. Die Gewichte, die zum Merkmalsvektor $\mathbf{x}^{(l)}(n)$ gehören, sollen daher ebenfalls nicht für die Auswertung berücksichtigt werden. Bei der Verwendung der Funktion Φ in (5.5) kann dieses erreicht werden, indem der Merkmalsvektor für den entsprechenden Sensor auf $\mathbf{0}$ gesetzt wird

$$\mathbf{x}^{(m)}(n) = \leftarrow \mathbf{0}, \tag{5.11}$$

dadurch gehen die entsprechenden Gewichte nicht in die Bestimmung der Farbe $s(n)$ ein. Zur Bestimmung des Zustands werden die übrigen Sensoren verwendet. Dieses bedeutet zwar, dass weniger Informationen zur Bestimmung des Zustandes genutzt werden, dennoch ist dieses für gewöhnlich einem kompletten Ausfall des Systems vorzuziehen.

5.3 Diskussion der bedingten Zufallsfelder zur Ereignisdetektion

In diesem Kapitel wurden die Anpassungen des CRFs zur Ereignisdetektion besprochen. Der wichtigste Punkt ist die Integration eines Ereigniszustands in das Modell. Für diesen müssen keine Beispiele in der Trainingsmenge vorhanden sein, es wird ausschließlich über die Abwesenheit des Normalfalls definiert.

Neben diesem Ereigniszustand ist die Interpretation der erzeugten Zustandssequenz eine weitere wesentliche Anpassung. Diese klassifiziert letztendlich ausschließlich zwischen Normalfall und Ereignis. Der Normalfall wird in mehrere Unterfälle aufgeteilt. Diese Aufteilung dient dazu, eine komplexere Struktur des Normalfalls, insbesondere in zeitlicher Hinsicht, zu erfassen. Dadurch können Ereignisse eventuell genauer vom Normalfall separiert werden, zudem ermöglicht dieses die Interpretation von langen Segmenten. Hier zeigt sich die Fähigkeit des CRFs als Übersetzungsschicht zwischen den oft sehr komplexen, realen Messwerten und einen für Algorithmen leichter verständlichen Kontext, also die Verteilung der Farben in der Zustandssequenz.

6 Experimente bezüglich der bedingten Zufallsfelder

In diesem Kapitel werden Experimente beschrieben, die unterschiedliche Aspekte der zur Ereignisdetektion entwickelten Algorithmen betrachten. Sie sind in mehreren wissenschaftlichen Arbeiten veröffentlicht worden. Die Experimente werden hier in chronologischer Reihenfolge ihrer Veröffentlichung vorgestellt.

Als erstes wird ein Beispiel für die Ereignisdetektion mittels eines CRFs vorgestellt [51]. Hierbei wird die Ereignisdetektion verwendet, um Kontrastmittelinjektionen innerhalb eines chirurgischen Eingriffs zu detektieren. Dieses Modell wird mit einem Verfahren verglichen, das zuvor speziell für diese Aufgabe entwickelt wurde.

Das zweite Experiment, das hier vorgestellt wird, zeigt das blinde Trainingsverfahren am Beispiel von akustischen Signalen [50]. Die Eingabe ist dabei ein Audiosignal, dafür wird eine Zustandssequenz gebildet, die das eingehende Signal beschreibt. Dadurch wird gezeigt, wie die Einteilung eines Signals in einzelne Gruppen vorgenommen wird, ein wichtiger Aspekt für die später betrachtete Einteilung des Normalfalls.

Das dritte Experiment zeigt die Möglichkeit einer Ereignisdetektion bei der Analyse von Videosequenzen [49]. Hierbei wird neben der eigentlichen Analyse ebenfalls das blinde Training von CRFs verwendet und somit die Kombination dieser beiden Methoden in einem praktisch relevanten Beispiel gezeigt.

Das letzte Experiment zeigt die Möglichkeit der Ereignisdetektion mittels CRFs anhand von Sensornetzwerken [48]. Hiermit wird gezeigt, wie CRFs zur Ereignisdetektion eingesetzt werden können, um sehr unterschiedliche Sensoren zu kombinieren. Ferner wird gezeigt, dass die Analyse auch dann nicht abbricht, wenn das Netzwerk zum Teil ausfällt. Dadurch wird insbesondere die Praxistauglichkeit dieses Verfahrens unterstrichen.

6.1 Ereignisdetektion zur Detektion von Kontrastmitteln

Dieses Experiment befasst sich mit der Detektion eines Kontrastmittels bei einer chirurgischen Operation. Diese Operation kann unter Umständen bedeutend für die Überlebenswahrscheinlichkeiten von Herzpatienten sein, daher handelt es sich um ein praktisch relevantes Problem. Das analytische Verfahren basiert auf der Methode der CRFs zur Ereignisdetektion, die in Kapitel 5 erläutert wurde. Zum Training wurde das Verfahren verwendet, das auf einer Strafmatrix beruht und somit einen neuen, vorher nicht beobachteten Ereigniszustand erzeugt. Das Experiment dient insbesondere zum Darstellen der Anwendbarkeit dieser Methode zur Ereignisdetektion. Dafür wird das CRF auf ein Signal trainiert, das aus echten Daten extrahiert wird und ein praktisch relevantes Problem beschreibt. Auch handelt es sich hierbei um ein Training mittels eines überwachten Verfahrens, das heißt, zum Training wird eine Zustandssequenz verwendet. Diese wird ebenfalls aus Trainingsdaten generiert.

Das Problem wurde bereits in [13–15] beschrieben. Bei der Behandlung handelt es sich um die perkutane transluminale Koronarangioplastie (PTCA), ein Verfahren aus der Kardiologie. Hierbei werden Verengungen von Koronargefäßen (Koronarstenosen), die mit Herzinfarkten in Zusammenhang stehen, beseitigt, indem ein Ballonkatheter an die entsprechende Position geführt und mit Druck das Gefäß erweitert wird. Um die korrekte Position zu finden, wird diese Operation mit Hilfe von ständigen Röntgenaufnahmen durchgeführt. Da Arterien in Röntgenaufnahmen nicht deutlich zu sehen sind, wird ein Kontrastmittel zugeführt. Um die korrekte Dosierung zu ermöglichen, wurde in [13–15] ein Verfahren entwickelt, mit dem in den Aufnahmen das Kontrastmittel detektiert wird, sodass die Zufuhr automatisch reguliert werden kann.

Der erste Schritt zur Detektion des Kontrastmittels ist die Bestimmung eines Merkmals für das Kontrastmittel. Hierbei wird ausgenutzt, dass die Gefäße ohne Kontrastmittel in Röntgenaufnahmen nur schwer zu sehen sind; sind sie deutlich sichtbar, ist das Kontrastmittel im sichtbaren Bereich. Dadurch ist die Sichtbarkeit von Arterien ein eindeutiges Merkmal.

Zuerst werden die Gefäße mittels eines Top-Hat-Filters [13, 19, 60] aus einer Aufnahme entfernt. Hierbei wird zuerst ein lokaler Maximumfilter auf die Grauwertbilder der Röntgenaufnahme angewendet, worauf anschließend ein lokaler Minimumfilter verwendet wird. Dieses Bild, das demnach dem Hintergrund entspricht, wird von der ursprünglichen Aufnahme abgezogen; das Differenzbild enthält somit die gesuchten

Gefäße. Ist das Kontrastmittel im sichtbaren Bereich, so sind in den Aufnahmen deutlich dunklere Werte zu erwarten, da das Kontrastmittel deutlich stärker absorbiert als das umliegende Gewebe. Das bedeutet, dass sich die Verteilung der Grauwerte in den Aufnahmen verändert. Als Maß hierfür wird das 98%-Perzentil verwendet, da durch die Bildung der Differenzbilder sehr niedrige Grauwerte im Röntgenbild hohen Werten im Bild mit entferntem Hintergrund entsprechen. Das Ergebnis ist demnach ein Wert $y_0(n)$ pro Bild der Sequenz. Das Signal y_0 wird anschließend gefiltert, um Rauschen zu reduzieren. Dafür wird das Signal y rekursiv definiert mit

$$y(n) = f(n) \cdot y_0(n) + (1 - f(n))y(n-1), \tag{6.1}$$

wobei $f(n)$ definiert ist durch

$$f(n) = \begin{cases} c \in [0,1) & \text{falls } y_0(n) - y_0(n-1) \leq 3\hat{\sigma} \\ 1 & \text{sonst,} \end{cases}$$

$\hat{\sigma}$ wird mittels der ersten Messwerte geschätzt.

6.1.1 Detektion des Kontrastmittels als Problem der Ereignisdetektion

Unter idealen Bedingungen steigt der Wert $y(n)$ deutlich, wenn das Kontrastmittel im sichtbaren Bereich ist. In diesem Fall genügt ein Schwellwert, um die Anwesenheit des Kontrastmittels zu detektieren.

Allerdings haben die Experimente auch gezeigt, dass dieses Verfahren anfällig für die Anwesenheit der Injektionsnadel für das Kontrastmittel im sichtbaren Bereich ist, und dem gewählten Ausschnitt und Einstellungen des Aufnahmegeräts, also den registrierten Grauwerten des Bildes. Verletzen die Umstände diese Annahmen, so kann es vorkommen, dass das Auftreten des Kontrastmittels nicht zum korrekten Zeitpunkt detektiert wird.

Da das Kontrastmittel direkten Einfluss auf die gemessene Größe hat, kann weiterhin vorausgesetzt werden, dass die statistischen Eigenschaften sich in dem Signal y messen lassen, das heißt, dass eine Veränderung des Signals stattfindet, auch wenn die Art dieser Veränderung vorher nicht bekannt ist. Dadurch lassen sich Methoden der Ereignisdetektion auf dieses Problem anwenden: Die Abwesenheit des Kontrastmittels wird als Normalfall angenommen, hierauf wird das Modell trainiert. Das Ereignis ist das Auftreten des Kontrastmittels im sichtbaren Bereich, was eine beliebige Veränderung

des Signals zur Folge hat. Damit ist das Problem in den Kontext der Ereignisdetektion überführt. Der praktisch relevante Anteil des folgenden Experiments ist demnach, in den Fällen, in denen das Kontrastmittel nicht mittels des separaten Verfahrens detektiert werden kann, das Ereignis zu erkennen, zugleich bei erfolgreichen Detektionen vergleichbare Ergebnisse zum bekannten Verfahren zu liefern.

6.1.2 Transformation

Es kann angenommen werden, dass das Auftreten des Kontrastmittels die Eigenschaften des Signals verändert. Da es sich um eine Operation am Herzen handelt, sind die Aufnahmen auch abhängig vom Herzschlag, das heißt, der beobachtete Bereich befindet sich in Bewegung. Dadurch existieren mehrere unterschiedliche Zustände. Hierfür wird das Markov-Modell aufgestellt: Jeder Zustand des Modells beschreibt hierbei einen Zustand, der durch die Aufnahme registriert werden kann. Diese Zustände unterscheiden sich durch Eigenschaften im Signal. Diese Eigenschaften werden aus dem Signal gemessen und zur Bestimmung des Zustands genutzt.

Die Eigenschaften, die zur Bestimmung des Zustands des Modells genutzt werden, beschreiben das Signal über mehrere Messwerte hinweg, sind demnach lokale Eigenschaften. Diese Eigenschaften umfassen den Mittelwert über unterschiedlich lange Segmente, Krümmung und Steigung. Diese Werte werden in einen Merkmalsvektor zusammengefasst. Ferner wird berücksichtigt, dass das Signal unterschiedlich stark rauschen kann und demnach Abschnitte mit Ausreißern weniger beachtet werden als solche ohne Ausreißer.

6.1.2.1 Mittelwert

Zwei Werte des Merkmalsvektors messen die Veränderung des Mittelwertes. Hierfür wird ein schnell und ein langsam adaptierender Mittelwert genommen, mit dem ein Wert $y(n)$ verglichen wird.

Seien $\alpha, \beta \in (0,1)$ sowie $T \in \mathbb{N}$. Dann sind der schnell adaptierende Mittelwert $\nu(n)$ und der langsam adaptierende Mittelwert $\mu(n)$ rekursiv definiert durch [51]

$$\mu(n) = \alpha\mu(n-1) + (1-\alpha)\sum_{m=1}^{T} y(n+m),$$
$$\nu(n) = \beta\nu(n-1) + (1-\beta)\mu(n-1).$$

Als Maße für die Abweichung von diesen Werten wird

$$d_n(m) = y(m) - \mu(n),$$
$$\tilde{d_n}(m) = y(m) - \nu(n)$$

verwendet. Um diese als Elemente für den Merkmalsvektor zu nutzen, werden sie mittels

$$\xi_1(n) = 10 \cdot \left(1 - \sqrt{\frac{\sum_{m=0}^{T-1} d_n(m+n)^2}{2 \cdot T \cdot \hat{\sigma}^2}}\right) \tag{6.2}$$

$$\xi_2(n) = 10 \cdot \left(1 - \sqrt{\frac{\sum_{m=0}^{T-1} \tilde{d_n}(m+n)^2}{5 \cdot T \cdot \hat{\sigma}^2}}\right) \tag{6.3}$$

in zwei verwendbare Werte transformiert.

6.1.2.2 Krümmung

Die Krümmung wird für einen Zustand explizit beschrieben. Im Detail bedeutet das, dass für einen Abschnitt der Länge T die T Werte

$$\xi_{2+k+1}(n) = y(n+k) - \frac{1}{T}\sum_{m=0}^{T-1} y(n+m) \tag{6.4}$$

als Merkmale verwendet werden.

6.1.2.3 Steigung

Zuletzt werden als weitere Merkmale die Steigung eines Segments verwendet. Unter der Steigung eines Segments wird hier verstanden, dass eine Gerade für die Messwerte nach Prinzip des kleinsten quadratischen Fehlers angepasst wird. Deren Steigung wird als Information über das Segment verwendet. Diese lässt sich beschreiben als $\hat{b}(n)$ mit

$$\hat{b}(n) = \arg\min_b \sum_{m=0}^{T-1} \left(y(n+m) - y(n) - b \cdot m\right)^2.$$

Um dieses Merkmal in dem Modell zu verwenden, wird es mit $L \in \mathbb{N}$ Teststeigungen verglichen. Dafür wird zunächst b_{min} als die in einer Trainingsmenge minimal vorhan-

dene Steigung respektive b_{max} als die maximale vorhandene Steigung gemessen. Eine Teststeigung ist w_l für $l = 1, 2, \ldots, L$

$$w_l = \frac{L-l}{L-1} b_{min} + \frac{l-1}{L-1} b_{max}.$$

Die dazugehörige Testfunktion ist

$$\xi_{2+T+l} = 1 - \left(\frac{\hat{b}(n) - w_l}{w_2 - w_1} \right) \tag{6.5}$$

für $l = 1, 2, \ldots, L$. Hiermit sind demnach $2+T+L$ Merkmale in jedem Merkmalsvektor enthalten. Dieser ist

$$\xi(n) = [\xi_l(n)]_{l=1}^{2+T+L}. \tag{6.6}$$

6.1.2.4 Ausreißerbehandlung

Das Signal des 98%-Perzentils beinhaltet aufgrund der Aufnahmemethode der Röntgenbilder einige Ausreißer. Als Ausreißer wird hier ein Messwert verstanden, der gegenüber seinem Vorgänger sowie Nachfolger einen deutlich abweichenden Wert annimmt. Da hierdurch die Variation innerhalb eines Segments stark erhöht wird, wird diese Eigenschaft zur Reduzierung des Einflusses der Ausreißer verwendet.

Sei $v(n)$ mit

$$v(n) = \sum_{m=1}^{T-1} \left(y(n+m) - y(n+m-1) \right)^2$$

die nichtnormalisierte lokale Variation eines Segments. Dann sind die Merkmalsvektoren, die zur Ereignisdetektion in diesem Verfahren verwendet werden, $\mathbf{x}(n)$ mit

$$\mathbf{x}(n) = \frac{1}{v(n)} \xi(n). \tag{6.7}$$

Dadurch haben Segmente mit Ausreißern einen geringeren Einfluss auf die Klassifikation als solche ohne Ausreißer; gleichzeitig wird die Information des Segments nicht vollständig ignoriert. Da zudem eine Markov-Kette bestimmt wird, also Nachbarschaftsbeziehungen bei der Klassifikation berücksichtigt werden und somit die fehlenden

Informationen dieses Segments durch die direkten Nachbarn ergänzt werden, ist dieses Verfahren stabil gegenüber Ausreißern mit den hier verwendeten Merkmalen.

6.1.3 Training und Modellbeschreibung

Das Markov-Modell, das hier verwendet wird, ist eine der sequentiellen Markov-Ketten, wie sie im Rahmen dieser Arbeit entwickelt und in Abschnitt 4.6 behandelt wurden. Dadurch ist eine echtzeitfähige Auswertung gegeben. Die Merkmale sind derart gewählt, dass sie mittels einer einfachen Funktion Φ in einem CRF verwendet werden können. Diese entspricht Gleichung (5.5), also

$$\Phi(s(n-1), s(n), \mathbf{x}(n)) = \begin{bmatrix} [[\![s(n) = \zeta_k]\!] \cdot \mathbf{x}(n)]_{k=1}^{K} \\ \left[[[\![s(n-1) = \zeta_j]\!] \cdot [\![s(n) = \zeta_k]\!]]_{k=1}^{K} \right]_{j=1}^{K} \\ [\![s(n) = \zeta_0]\!] \cdot \mathbf{x}(n) \end{bmatrix}.$$

Das Training für das CRF ist in diesem Experiment ein überwachter Algorithmus. Demnach wird das CRF mittels einer Trainingsmenge und einer dazugehörigen Sequenz an Zuständen trainiert. Diese Trainingssequenz an Zuständen wird mittels eines einfachen Verfahrens bestimmt, das jedoch nur dazu geeignet ist, Klassen des Normalfalls zu beschreiben. Direkt zur Ereignisdetektion ist es nicht geeignet.

Sei hierfür y_{min} der minimal gemessene Wert innerhalb der Trainingsmenge und y_{max} der maximal gemessene Wert. Sei ferner

$$a_q = \frac{K-q+1}{K} y_{min} + \frac{q-1}{K} y_{max} \tag{6.8}$$

für $q = 1, 2, \ldots, K+1$. Dann ist für die Trainingssequenz $s(n) = \zeta_k$, wenn gilt

$$a_q \leq \frac{1}{T} \sum_{m=0}^{T-1} y(n+m) < a_{q+1}, \tag{6.9}$$

das bedeutet, dass der Zustand eines Elements der Trainingssequenz vom Mittelwert eines entsprechenden Segments bestimmt wird.

Die Trainingsmenge an Merkmalsvektoren entspricht Messungen in den ersten Sekunden, bevor das Kontrastmittel in den sichtbaren Bereich dringt. Dieser Moment wurde manuell festgelegt. Das Training erfolgt mittels einer Strafmatrix $\mathbf{D}$, wie sie in (5.10) beschrieben wird. Dadurch wird ein Zustand für das Ereignis mit trainiert, auch wenn keine Merkmalsvektoren, die das Ereignis beschreiben, in der Trainingsmenge enthalten sind.

Als Länge der Segmente, aus denen die Merkmale errechnet werden, wurde $T = 36$ gewählt, dieses entspricht 36 Bildern. Ferner werden $L = 3$ Teststeigungen für die entsprechenden Merkmale verwendet, sodass insgesamt jeder Merkmalsvektor $\mathbf{x}(n)$ 41 Merkmale enthält. Für ein Segment des Markov-Modells zur sequentiellen Auswertung werden zehn Merkmalsvektoren betrachtet, in jeder der Iterationen werden acht Zustände bestimmt. Jedem Merkmalsvektor wird ein Zustand zugeordnet, somit beträgt die Überlappung, wie sie in der sequentiellen Auswertung verwendet wird, zwei Zustände.

6.1.3.1 Entscheidung mittels kumulierter Summe (CUSUM-Test)

Um Fehlklassifikationen zu vermeiden, wird die Entscheidung, ob ein Ereignis stattgefunden hat, mittels einer kumulierten Summe (CUSUM-Test) [4] gefällt. Die Teststatistik ist $c(n)$ mit

$$c(n) = c(n) - 1 + P(s(n) = \zeta_0) - c_0, \tag{6.10}$$

wobei $P(s(n) = \zeta_0)$ die Wahrscheinlichkeit für die Beobachtung eines Ereignisses ist; an dieser Stelle wurde aus Gründen der Übersichtlichkeit die Abhängigkeit bezüglich der Nachbarn und der Abhängigkeit vom Merkmalsvektor in der Notation verzichtet. Ein Ereignis ist detektiert, falls $c(n) \geq 1$. Der Parameter c_0 ist derart gewählt, dass dieses nicht in der Trainingsmenge eintritt.

6.1.4 Ergebnisse

Der Algorithmus wurde auf neun Sequenzen angewendet. Bei diesen tritt insbesondere hervor, dass die Voraussetzungen des ursprünglichen Algorithmus [13–15] verletzt wurden. In acht der Sequenzen wurde das Kontrastmittel detektiert, wobei es zum Teil nicht mit dem ursprünglichen Algorithmus erkannt werden konnte. Die Signale und die dazugehörigen Klassifikationen sind in Abbildung 6.1 zu sehen. In einer der Sequenzer n konnte weder das klassische noch das vorgestellte Verfahren das Ereignis detektieren (nicht dargestellt).

Durch die ausreichend genaue Klassifikation des Kontrastmittels ist die Detektion von Ereignissen mittels der sequentiellen Auswertung von Segmenten eines CRFs, das für die Ereignisdetektion konzipiert wurde, anwendbar. Diese Methode zeigt sich robuster gegenüber Fehlern bei der Aufnahme als das Standardverfahren.

6.2 Blindes Training eines bedingten Markov-Zufallsfeldes

In diesem Experiment ist das zentrale Problem, ein vorhandenes Signal mittels eines CRF-Trainings in mehrere Gruppierungen, genannt Cluster [3, 11, 68], aufzuteilen. Dieses Problem ist auch für die Ereignisdetektion wichtig, da sich damit der Normalfall genauer strukturieren lässt. Das hierzu gehörige Experiment wurde in [50] veröffentlicht. Es bildet die Grundlage für das in [49] veröffentlichte Verfahren zur Ereignisdetektion.

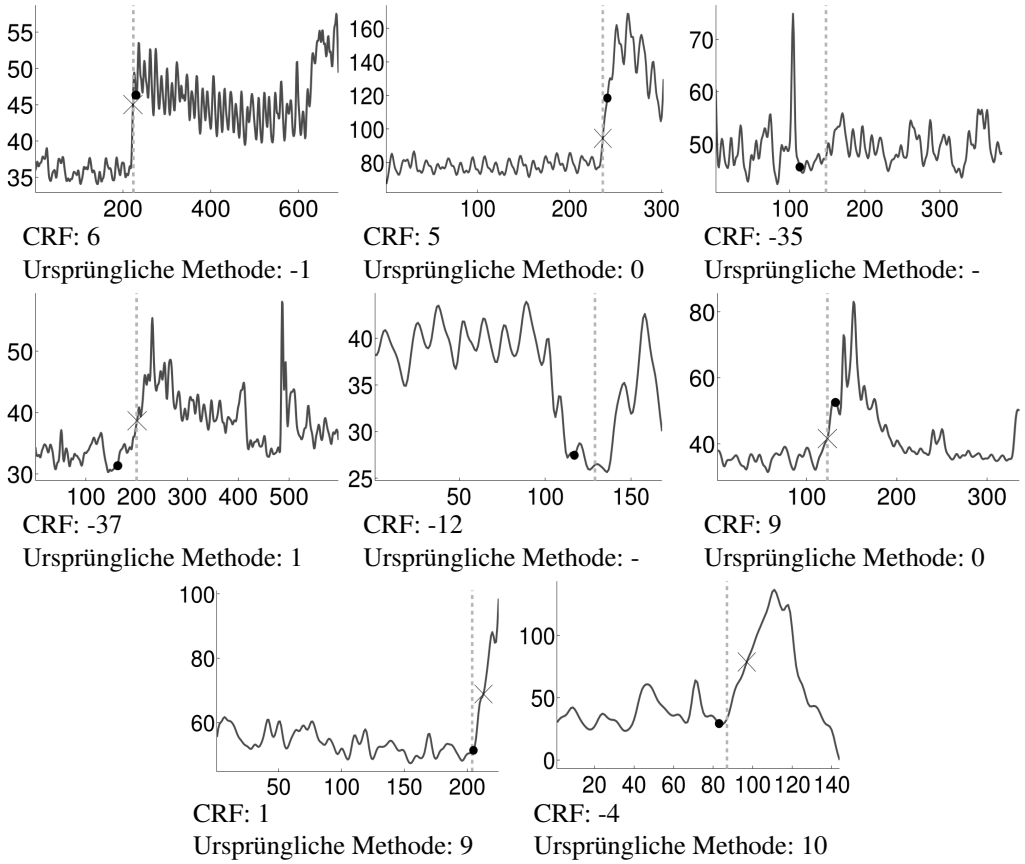


Abbildung 6.1: Signale des Kontrastmittels in unterschiedlichen Aufnahmen. Die Detektion mittels eines CRFs ist durch einen Punkt markiert, zum Vergleich sind ebenfalls die ursprüngliche Methode in [13] (Kreuz) und eine manuelle Auswertung (senkrechte Linie) angegeben. In zwei Fällen detektierte des CRF das Ereignis, bei dem die ursprüngliche Methode es nicht gemessen hat, bei den anderen Beispielen zeigten sich beide Methoden als anwendbar.

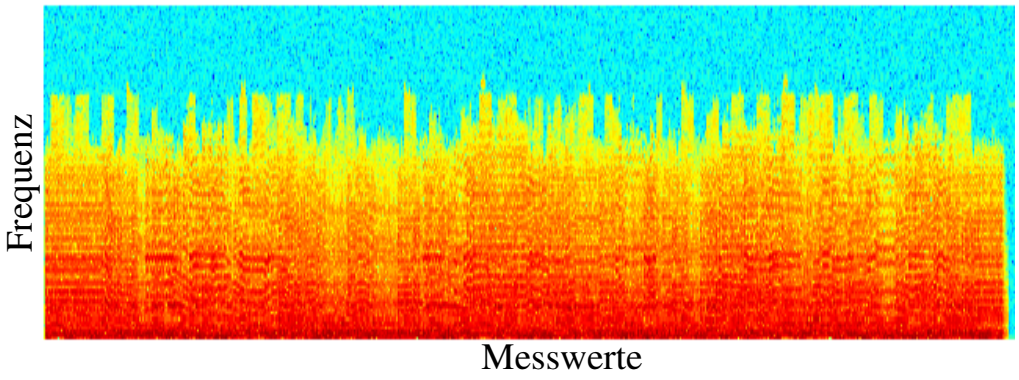


Abbildung 6.2: Spektrogramm des verwendeten Audiosignals.

Die Verwendung eines Clusteralgorithmus ist eine praktische Methode der Vorverarbeitung für unterschiedliche Probleme. Markov-Modelle sind hierbei ein Standardverfahren [80]. Neu ist die Verwendung von CRFs. Als datengetriebene Modelle sind diese für sehr unterschiedliche Daten, auch ohne Modellierung von Verteilungen, geeignet. Dadurch ist dieses Verfahren auch für andere Probleme als die Ereignisdetektion von Interesse.

6.2.1 Datenbasis

Es können viele unterschiedliche zeitabhängige Signale mit dem hier vorgestellten Verfahren in eine feste Anzahl an Clustern eingeteilt werden. Hierbei ist die Merkmalsextraktion wichtig, um das Signal nach erwünschten Eigenschaften zu gruppieren. Um in diesem Experiment die Merkmalsextraktion einfach zu halten, wurde das Spektrogramm eines Audiosignals verwendet. Das Audiosignal ist eine Interpretation von Bachs Werk "Air" mit einer Samplingrate von 44100Hz, siehe Abbildung 6.2. Das Spektrogramm wurde mit einem Hamming-Fenster [57] mit einer Weite von 256 Werten (Samples) und einer Überschneidung von 128 Werten berechnet. Anschließend wird jedes Band normiert, sodass im Gesamtsignal in jedem Band dieselbe Energie enthalten ist. Das CRF wird auf die ersten 20% der so berechneten Merkmale trainiert. Mit dem trainierten CRF wird anschließend das gesamte Signal in Cluster eingeteilt. Zur Beurteilung des Erfolges wird die Sequenzentropie gemessen: Je höher diese ist, desto besser ist das Verfahren zum Segmentieren dieses Signals geeignet. Ferner wird die Eindeutigkeit, mit der ein Messwert einem Cluster zugeordnet wird, berücksichtigt.

6.2.2 Interpretation als Cluster-Problem

In dem Experiment geht es um die Frage, mittels eines CRFs die Struktur eines Audiosignals zu analysieren. Dabei werden Abschnitte mit ähnlichen Eigenschaften in ein Cluster eingeteilt. Festgelegt wird hierfür die Anzahl an möglichen Clustern. Dieses ist der einzige vorher festgesetzte Parameter.

Insbesondere wird eine Zustandssequenz $\hat{\mathbf{S}}$ zu einer Merkmalssequenz $\mathbf{X}$ gleicher Länge erstellt. Das heißt, die Sequenz wird geschätzt nach

$$\hat{\mathbf{S}} = \arg\max_{\mathbf{S}} p(\mathbf{S}|\mathbf{X};\lambda), \tag{6.11}$$

wobei λ die trainierten Parameter des CRFs sind. Zwei Merkmale $\mathbf{x}(n)$ und $\mathbf{x}(m)$ gehören genau dann zum selben Cluster, wenn gilt $\hat{s}(n) = \hat{s}(m)$.

Zusätzlich ist erwünscht, dass die Cluster vergleichbar groß sind, das heißt, dass die Anzahl der Merkmale, die zu einem Cluster gezählt werden, nicht zu sehr von der Anzahl von Merkmalen in anderen Clustern abweichen. Eine exakte Gleichheit an Merkmalen pro Cluster ist zu streng, zum Beispiel wäre es nur bei bestimmten Anzahlen von Merkmalen möglich, diese Bedingung zu erfüllen. Deswegen wird diese geringere Bedingung gestellt.

Zur Bewertung der erstellten Cluster wird ein Maß verwendet, wie es in [80] ebenfalls genutzt wird. Angepasst wird dieses an die besonderen Eigenschaften eines CRFs. Ferner wird die Entropie der Zustände berücksichtigt sowie das Maß bezüglich der Länge der Sequenzen normalisiert. Dadurch wird ein Maß für die Forderung erzeugt, dass alle Cluster vergleichbar groß sein sollen. Das verwendete Maß zur Bestimmung der Qualität des Clusters ist $\nu(\hat{\mathbf{S}},\lambda)$ mit

$$\nu(\hat{\mathbf{S}},\lambda) = H(\hat{\mathbf{S}}) \cdot \left(\prod_{n=1}^{N} K \cdot p(\hat{s}(n)|\mathbf{x}(n);\lambda) \right)^{\frac{1}{N}}, \tag{6.12}$$

wobei N die Anzahl der Merkmalsvektoren und somit die Länge der Zustandssequenz ist, K ist die Anzahl an Clustern und $H(\hat{\mathbf{S}})$ ist die Sequenzentropie

$$H(\hat{\mathbf{S}}) = -\sum_{k=1}^{K} h(\hat{\mathbf{S}},\zeta_k) \cdot \log(h(\hat{\mathbf{S}},\zeta_k)),$$

$$h(\hat{\mathbf{S}},\zeta_k) = \frac{1}{N} \sum_{n=1}^{N} [\![\hat{s}(n) = \zeta_k]\!].$$

Dieses Maß ermöglicht es, eine geschätzte Sequenz zu bewerten. Es kombiniert die Eindeutigkeit, mit der ein Zustand gemessen wird, mit der Entropie, also einem Maß der Auftrittshäufigkeit aller Merkmale. Eine gute Einteilung der Merkmale in die Cluster ist dann erreicht, wenn die Eindeutigkeit jeder Einteilung hoch ist und jeder mögliche Zustand ζ_k für $k = 1, 2, \ldots, K$ gleich oft in der geschätzten Sequenz vorkommt. In diesem Fall ist ebenfalls das vorgestellte Maß hoch.

6.2.2.1 Experimente

Um den Algorithmus zu bewerten, werden unterschiedliche Signale nach dem Training verwendet. Das Training wird auf die frühen Merkmalsvektoren des genannten Signals ausgeführt. Anschließend wird in einem ersten Experiment eine Zustandssequenz für die gesamte Sequenz erstellt.

Im zweiten Experiment wird das Signal an einer Position durch eine Sirene mit geringerer Lautstärke als das reine Signal gestört. Ebenfalls wird hierfür eine Zustandssequenz und das entsprechende Maß berechnet.

Im dritten Experiment wird zum Vergleich ein weißes Rauschen verwendet. Es wird die gleiche Anzahl von Merkmalsvektoren erstellt, und hierfür eine Zustandssequenz ermittelt. Vergleicht man ausschließlich die Entropie $H(\hat{S})$, so ist dieser Wert in diesem Experiment nahe dem theoretischen Maximum, während die Entropie bei anderen Signalen diese deutlich geringer ist. Diese dient als Vergleich zu den anderen beiden Experimenten. Alle Experimente werden fünf bis zehn mal wiederholt, um die Ergebnisse zu sichern. Die in Tabelle 6.1 angegebenen Werte sind über alle Versuche gemittelt. Die Unterschiede über die Wiederholungen des Experiments waren gering, das heißt, die Werte unterschieden sich um weniger als die kleinste angegebene Dezimalstelle (weniger als 10^{-5}). Dadurch sind die Ergebnisse verlässlich bezüglich ihrer Aussagekraft.

6.2.3 Ergebnisse

In Tabelle 6.1 sind die Ergebnisse für das Clusterexperiment zusammengefasst. Hierbei handelt es sich um das Qualitätsmaß $\nu(\hat{\mathbf{S}}, \lambda)$, wie es in (6.12) beschrieben wird. Zum Vergleich ist ebenfalls das Maß für ein weißes Rauschen angegeben sowie eines Signals, bei der die Störung eingefügt wurde.

Die Zustandsentropie $H(\hat{\mathbf{S}})$ ist bei dem Experiment, bei dem ausschließlich weißes Rauschen verwendet wurde, maximal. Dieses ist durch die Gleichheit der A-priori-Wahrscheinlichkeiten der Zustände gegeben. Die Unterschiede zu den Experimenten mit

Tabelle 6.1: Klassifikationsmaß ν für das Clusterexperiment, wie in (6.12) beschrieben. Angegeben sind für die drei Testsignale die Maße für unterschiedliche Anzahl von Zuständen.

	$K = 2$	$K = 3$	$K = 4$	$K = 6$
Signal ohne Rauschen	1.8443	2.6956	3.4150	4.9168
Gestörtes Signal	1.8583	2.6843	3.3772	4.9089
Rauschen	1.5508	1.8968	2.1725	2.5740

dem reinen Signal sowie mit dem gestörten Signal ergeben sich aus der Eindeutigkeit, mit der die Zustände bestimmt werden können; dieses bedeutet, dass die Zustände mit dem trainierten CRF gut identifiziert werden konnten.

Mit steigender Anzahl von Zuständen erhöht sich der Abstand zwischen den Experimenten mit dem klaren respektive gestörten Signal. Das bedeutet, dass die steigende Anzahl der Zustände eine höhere Genauigkeit der Klassifikation zulassen.

Der Unterschied zwischen dem klaren und dem gestörten Signal weist darauf hin, dass die Störung deutlich gemessen werden konnte. Die Störung ist einige Sekunden lang und dadurch deutlich kürzer als das eigentliche Testsignal. Diese Eigenschaft erlaubt den Rückschluss, dass Verfahren, die auf dem blinden Training basieren, zur Ereignisdetektion genutzt werden können, da der unbekannte Anteil des Signals diese Abweichung verursacht. Dieses ermutigende Ergebnis führte zu den nachfolgenden Experimenten.

6.3 Videoüberwachungsbeispiel zur Ereignisdetektion

Dieses Experiment wurde in [49] veröffentlicht. Das Szenario dieses Experiments ist ein klassisches Überwachungsszenario. Eine Kamera ist in einem zu beobachtenden Bereich fest installiert. In dem Sichtfeld werden unterschiedliche Aktionen durchgeführt.

Angenommen wird, dass in dem sichtbaren Bereich eine Person, das ist der Wachmann, einige Aktionen durchführen kann, die zu seiner Tätigkeit gehören. Dazu gehört, den sichtbaren Bereich zu durchqueren, in einen angrenzenden Raum zu treten oder durch ein Fenster in der Tür zu diesen Raum zu blicken. Die Aktionen des Wachmanns gelten als Normalfall.

Zudem wird eine weitere Person angenommen, der Einbrecher. Dieser hat Absichten, die sich von denen des Wachmanns unterscheiden. Es wird hier angenommen, dass der Einbrecher versucht, seine Aktionen zu verheimlichen; das bedeutet, dass er weder Objekte zeigt, die seine Tätigkeit verraten, noch wird er durch Kleidung oder ähnlich

offensichtliche Eigenschaften von dem Wachmann zu unterscheiden sein. Zudem wird er versuchen, seine Aktionen denen des Wachmanns anzupassen. Dieses wird durch den üblichen Aufbau der Kameras erleichtert: Damit diese einen großen Bereich abdecken, befindet sich die beobachtete Person oft zwischen Kamera und Objekt und verdeckt somit seine Aktionen.

Eine einfache Klassifizierung der Ereignisse mittels Objekt- oder Gesichtserkennung ist somit ausgeschlossen. Auch werden sich die Aktionen generell nur geringfügig vom Normalfall unterscheiden. Bei einem direkten Training ist somit eine hohe Falsch-negativ-Rate möglich, weswegen ein robusterer Ansatz in der Sequenzanalyse gewählt wird.

Es wird dabei ausgenutzt, dass die Aktionen sich auf längere Sicht zwangsläufig unterscheiden: Es kann angenommen werden, dass die Aktion, ein Schloss zu öffnen, üblicherweise mit dem Schlüssel schneller vollzogen ist als mit alternativen Werkzeug, ferner wird der Einbrecher versuchen, andere Aktionen kurz zu halten, um Begegnungen mit dem Wachmann zu vermeiden. Somit gibt es Abweichungen in der Dauer der einzelnen Aktionen. Dadurch kann mittels einer Klassifikation der Aktionen das Ereignis bestimmt werden, indem zunächst die übliche Dauer einer Aktion gemessen wird. Weicht eine Zeit ab, so wird ein Ereignis angenommen.

Der kritische Aspekt ist hier die Klassifikation der Aktionen. In diesem Experiment erfolgt diese mittels eines unüberwacht gelernten CRFs. Die Dauer der Aktionen wird mittels GMMs gemessen.

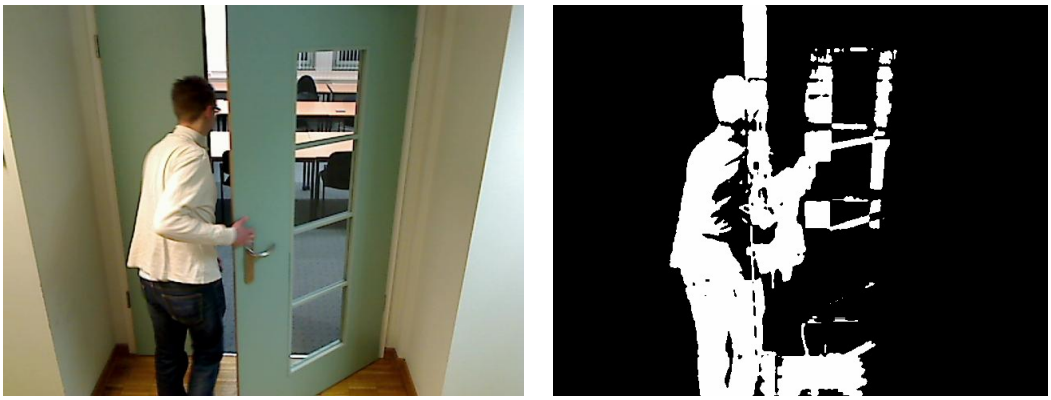
6.3.1 Merkmalsextraktion

Die Kamera nimmt Farbbilder mit 30 Bildern pro Sekunde auf. Aus diesen wird ein Vordergrundbild extrahiert. Dafür wird ein ähnlicher Algorithmus wie in [52] verwendet. Dieser Algorithmus wurde für Farbbilder angepasst.

Sei $\mathbf{F}_n$ mit $n = 0, 1, 2, \ldots$ ein aufgenommenes Bild mit der Auflösung $M_1 \times M_2$, also

$$\mathbf{F}_n \ : \ \{(i,j) | 1 \leq i \leq M_1, 1 \leq j \leq M_2\} \rightarrow [0,1]^3,$$

und $F_n(i,j)$ ist ein Pixel des Bildes $\mathbf{F}_n$ an der Position (i,j). Sei ferner in den ersten



(a) Aufgenommenes Bild (b) Vordergrund

Abbildung 6.3: Bild einer Videosequenz im Beispiel der Videoüberwachung (a) und extrahierte Merkmale (b). Pixel, die in (b) weiß dargestellt sind, wurden zum Vordergrund gehörig klassifiziert.

n_0 Bildern keine Person im sichtbaren Bereich. Dann wird das Hintergrundmodell initialisiert mit

$$\mathbf{B}_{n_0} = \frac{1}{n_0} \sum_{n=0}^{n_0} \mathbf{F}_n.$$

Sei ferner $T_{n_0}(i,j) = 1$ ein Pixel des Schwellwertbildes $\mathbf{T}_{n_0}$. Dann wird das Vordergrund-Bild $\mathbf{I}_n$ für $n > n_0$ berechnet mit

$$I_n(i,j) = [\![\|F_n(i,j) - B_{n-1}(i,j)\|_2 > T_{n-1}(i,j)]\!], \tag{6.13}$$

wobei $B_n(i,j)$ und $T_n(i,j)$ für $n > n_0$ rekursiv definiert sind mit

$$B_n(i,j) = \begin{cases} B_{n-1}(i,j) & \text{falls } I_n(ij) = 1 \\ \alpha F_n(i,j) + (1-\alpha) B_{n-1}(i,j) & \text{sonst,} \end{cases}$$

$$T_n(i,j) = \begin{cases} T_{n-1}(i,j) & \text{falls } I_n(ij) = 1 \\ \alpha(\|F_n(i,j) - B_{n-1}(i,j)\|_2 + T_{off}) & \\ \quad +(1-\alpha) T_{n-1}(i,j) & \text{sonst,} \end{cases}$$

wobei die Konstante $T_{off} = 0.25$ zur Rauschunterdrückung verwendet wird. Der Merkmalsvektor für dieses Experiment ist

$$\mathbf{x}(n) = \left[[I_n(i,j)]_{i=1}^{M_1} \right]_{j=1}^{M_2}. \tag{6.14}$$

Der Merkmalsvektor entspricht demnach der Entscheidung, ob ein entsprechendes Pixel zum Vordergrund gehört, siehe Abbildung 6.3. Das hier genutzte Verfahren führt nicht zu einer vollständigen Verfolgung von Objekten, es genügt allerdings den hier verwendeten Methoden. Dieser Algorithmus wurde ebenfalls im Rahmen dieser Arbeit entwickelt und in [49] veröffentlicht.

Es werden mehrere Videos mit einer üblichen Aktionsfolge aufgenommen. Dieses umfasst sowohl Aktionen des Normalalls als auch des Ereignisses. Das bedeutet, dass diese Videos immer damit beginnen, dass die Person den sichtbaren Bereich betritt, und enden, wenn sie diesen verlässt. Eine Handlung ist eine Verkettung von möglichen Handlungen innerhalb des observierten Bereiches. Das CRF wird auf 40% der Videos trainiert, die nur den Normalfall enthalten, wie es bei allen hiesigen Algorithmen üblich ist. Die Auswahl der Videos erfolgt zufällig.

6.3.2 Training

Das Training des CRFs ist das in Abschnitt 4.7.2 vorgestellte blinde Verfahren. Hierfür seien λ_k und λ_{jk} eine Zerlegung des Gewichtvektors λ wie in (5.7), das heißt, in λ_k sind die Parameter, die den Zustand ζ_k beschreiben, λ_{jk} bestimmt die Wahrscheinlichkeit, von Zustand ζ_j in den Zustand ζ_k zu wechseln. Sei ferner $M = M_1, \cdot M_2$. Dann werden die Nebenbedingungen

$$\sum_{i=1}^{M} \lambda_k(i) = 0, \qquad k = 1, 2, \ldots, K, \tag{6.15}$$

$$\sum_{i=1}^{M} (\lambda_k(i))^2 = 1, \qquad k = 1, 2, \ldots, K, \tag{6.16}$$

$$\sum_{k=1}^{K} \lambda_{jk} = 0, \qquad j = 1, 2, \ldots, K, \tag{6.17}$$

$$\sum_{k=1}^{K} (\lambda_{jk})^2 = 1, \qquad j = 1, 2, \ldots, K., \tag{6.18}$$

verwendet, um die Parameter für die Zustände zu kontrollieren. Dieses ermöglicht die praktische Durchführung des unüberwachten Trainings. Die Bedingung (6.15) verringert die Zahl der Iterationsschritte im Trainingsalgorithmus. Die Bedingung (6.16) kontrolliert die A-priori-Wahrscheinlichkeit der Zustände, das bedeutet, dass vor der Messung eines neuen Bildes ohne Berücksichtigung der Übergangswahrscheinlichkeiten alle Zustände gleich wahrscheinlich sind. Die Bedingungen (6.17) und (6.18) kontrollieren die Übergangswahrscheinlichkeiten und dadurch eine Vergleichbarkeit zwischen diesen und den Parametern der Zustände. Hiermit ist der M-Schritt des Trainings, wie er in Abschnitt 4.7.2 beschrieben wird, definiert.

Für den E-Schritt wird eine weitere Annahme getroffen. Diese ist, dass Aktionen in mehreren aufeinander folgenden Bildern beobachtet werden können. Da eine Zustandssequenz eine Abfolge von Aktionen beschreiben soll, wird auch angenommen, dass ein Segment gleicher Färbung länger als nur einen Zustand ist; das heißt, dass dieselbe Färbung in mehreren sukzessiven Zuständen beobachtet wird. Dieses wird durch eine einfache Methode erreicht. Angenommen, für einen im E-Schritt geschätzten Zustand $\hat{s}(n)$ gilt, dass $\hat{s}(n) \neq \hat{s}(n-1)$ und $\hat{s}(n) \neq \hat{s}(n+1)$. Dann wird $\hat{s}(n)$ auf $\hat{s}(n-1)$ gesetzt falls $p(\hat{s}(n) = \hat{s}(n-1)|\mathbf{X}) \geq p(\hat{s}(n) = \hat{s}(n+1)|\mathbf{X})$, andernfalls auf $\hat{s}(n+1)$, das heißt, ein isolierter Zustand bekommt die Färbung des Nachbarn mit der höheren Likelihood.

Mit diesen Bedingungen kann das CRF unüberwacht trainiert werden. Das Ergebnis ist ein CRF, das zu der aufgenommenen Merkmalssequenz eine Zustandssequenz erzeugt. Diese wird anschließend interpretiert, um Ereignisse zu messen.

6.3.3 Interpretation der blind gelernten Segmente zur Detektion von Einbrüchen

Eine Farbe des CRFs, das in diesem Experiment genutzt wird, kann mit einer (elementaren) Aktion gleichgesetzt werden. Aus diesen Aktionen setzt sich eine gesamte Sequenz zusammen. Als Ereignis wird angenommen, wenn eine klassifizierte Aktion länger oder kürzer dauert, als in den Trainingssequenzen. Hierfür werden die Zustandssequenzen für die Trainingssequenzen erstellt, indem eine Sequenz geschätzt wird, wobei jeder Knoten die Farbe mit der höchsten Likelihood erhält. Die Längen der Segmente gleicher Färbung werden ermittelt, für jede Färbung wird ein GMM mit drei Normalverteilungen trainiert.

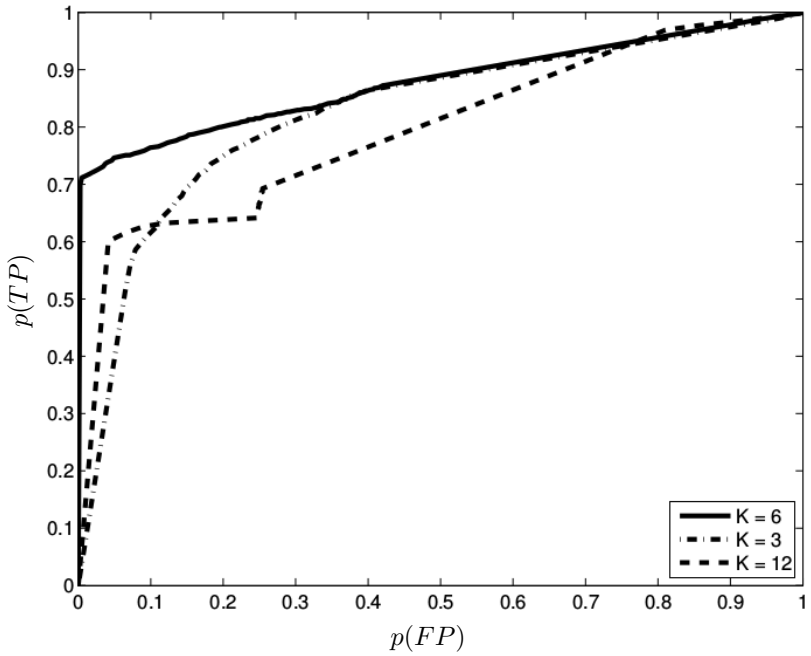


Abbildung 6.4: Grenzwertoptimierungskurven für das Experiment der Videoüberwachung [49]. Angegeben ist die unterschiedliche Anzahl von Farben K. Optimal ist in diesem Fall $K = 6$.

Bei einem neuen Video werden ebenfalls die Segmente gleicher Färbung bestimmt. Ist die Likelihood nach dem entsprechenden GMM gering, so wird angenommen, dass ein Ereignis beobachtet wurde.

6.3.4 Ergebnisse

Es wurden nur Videos, auf die nicht trainiert wurde, bei der Klassifikation berücksichtigt. Dieses gilt sowohl für Videos, die zu dem Normalfall gehören, als auch für Videos, in denen ein Ereignis eintritt. Beides wird von derselben Person durchgeführt, um Abweichungen in Größe, Farbe oder anderen Parametern zu vermeiden.

Da es nicht möglich ist, das Ereignis auf die tatsächlichen Bilder zu beschränken, wird nach konservativen Gesichtspunkten angenommen, dass jedes Bild eines Videos, das zum Ereignis gehört, als solches klassifiziert werden kann. In diesen Videos befinden sich auch Aktionen, die zum Normalfall gehören. Dieses hat Einfluss auf die Grenzwertoptimierungskurve, die in 6.4 angegeben ist. Es kann daher angenommen werden, dass die Ereignisse noch deutlicher detektiert werden können, als die Grenzwertoptimierungskurve darstellen kann.

Mit $K = 6$ Farben werden dennoch 71% der Bilder, die zum Ereignis gehören, als solche klassifiziert, wenn man eine Falsch-positivRate über alle Videosequenzen von 0% zur Basis nimmt; das bedeutet, dass jedes der Ereignis-Videos bei einer signifikant geringen Fehlerrate als solche klassifiziert werden konnten. Bei mehr oder weniger Farben verringert sich die Qualität der Klassifikation, sie ist jedoch stets hoch.

6.4 Analyse mit unvollständigen Datensätzen in inhomogenen Sensornetzwerken

Ein wesentlicher Vorteil von CRFs ist die Verwendungsmöglichkeit bei heterogenen Sensornetzwerken. Da CRFs datengetriebene Modelle sind, werden zunächst keine Annahmen über die Verteilung der Messwerte getroffen. Dadurch ist es möglich, sehr unterschiedliche Sensoren in einem geschlossenen Modell auszuwerten.

In diesem Experiment, das in [48] veröffentlicht wurde, werden Daten aus solchen Netzwerken ausgewertet. Hierbei handelt es sich um ein Problem der Umfeldüberwachung für einen umgebungsgestützten Haushalt. Ein Sensornetzwerk misst unterschiedliche Parameter, auf die eine Person des Haushalts Einfluss nimmt, zum Beispiel Temperatur und Luftfeuchtigkeit, oder die auf Aktionen der Person direkt hinweisen, wie Bewegungen von Einrichtungsgegenständen und Türen oder Bewegungssensoren. Weichen die Messungen von den trainierten Werten ab, so kann dieses auf eine pathologische Veränderung der beobachteten Person hinweisen. Dieses ist das Ereignis für das hier vorgestellte Experiment.

Da die Sensoren an sehr unterschiedlichen Positionen des überwachten Haushalts installiert werden, ist es nützlich, drahtlose Sensoren mit unabhängigen Energieversorgungen zu nutzen. Dadurch entsteht der Vorteil, dass bei Ausfällen Teile des Netzwerkes weiterhin zur Klassifikation genutzt werden können. Der zentrale Aspekt dieses Experiments ist es, zu zeigen, dass CRFs zur Ereignisdetektion in diesen Netzwerken verwendet werden können. Ferner wird gezeigt, dass sie ebenfalls genutzt werden können, wenn einige der Sensoren ausfallen. Dadurch sind CRFs gut geeignete Modelle für dieses Problem.

6.4.1 Aufbau eines drahtlosen Sensornetzes

Für dieses Experiment wurden Sensoren in einem Haushalt installiert. Diese messen Bewegungen von Möbeln wie einem Bett oder der Tür eines Schrankes respektive Kühl-

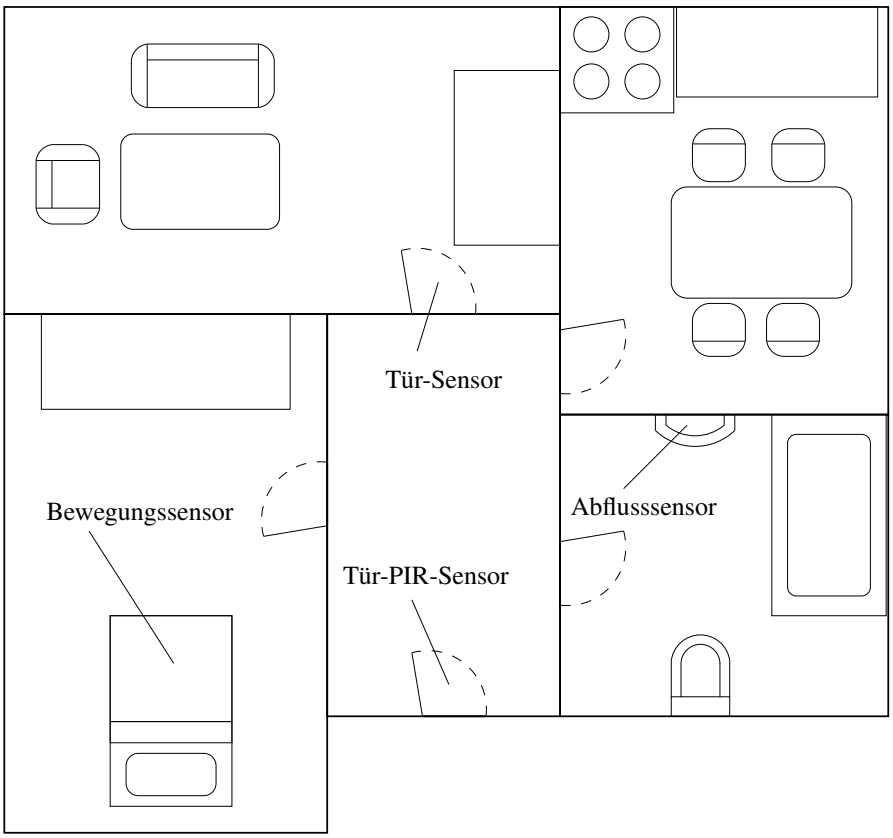


Abbildung 6.5: Stilistischer Aufbau einer typischen Wohnung mit entsprechenden Sensoren. Die Sensoren erfassen Temperatur und Luftfeuchtigkeit am Abfluss im Badezimmer sowie Bewegungen am Bett, einer oft genutzten Tür und der Eingangstür (PIR-Sensor: Passiv-Infrarot-Sensor).

schrankes, Türen der Wohnung, die häufig verwendet werden, sowie Bewegungen an einigen oft passierten Stellen. Ferner werden Parameter des Haushalts, insbesondere des Badezimmers, gemessen, so die Luftfeuchtigkeit und Temperatur, wie in Abbildung 6.5 angegeben. Die Sensoren sind drahtlos mit unabhängiger Stromversorgung und verbinden sich mit einer zentralen Station, die die gemessenen Werte an einen Server senden, der für mehrere Haushalte die Daten abspeichert.

Jeder Sensor misst unabhängig von den anderen Sensoren. In der Regel wird eine Messung pro Minute durchgeführt. Eine Ausnahme sind Bewegungssensoren, die zusätzlich bei Aktivierung einen Messwert senden. Dadurch erhält jede Messung jedes Sensors einen eigenen Zeitstempel. Ferner sind die Sensoren nicht voneinander abhängig, sodass sie nicht auf die Bereitschaft eines anderen Sensors warten müssen. Durch die unabhängige Stromversorgung ist des Weiteren die Sendeleistung begrenzt. Hierdurch und mitunter durch die unterschiedliche Baubeschaffung der Wohnung kann es vorkommen, dass die Signale nicht oder verspätet gesendet werden. Eine Echtzeit-Auswertung des Signals ist dadurch nicht möglich. In Abschnitt 6.4.2 werden die Daten in ein mehrdimensionales, synchrones zeitabhängiges Signal transformiert. Hierdurch können Verzögerungen und kürzere Ausfälle kompensiert werden.

Ein längerfristiger Ausfall eines Sensors tritt vor allem dann ein, wenn die Stromversorgung erschöpft ist. Auch bei schneller Reaktion können so Messwerte über mehrere Stunden ausbleiben. Dieses durch die hier durchgeführte Transformation zu kompensieren ist nicht praktikabel. Allerdings bieten CRFs die Möglichkeit, eine Klassifikation mit nur einem Teil der Sensoren durchzuführen.

6.4.2 Datenaufnahme und Datentransformation

Die Daten werden zunächst mit einem dazugehörigen Zeitstempel in einer zentralen Datenbank gespeichert. Die Daten sind über einen längeren Zeitraum abrufbar. Dadurch können Daten aus einem Zeitintervall zur Analyse genutzt werden.

Die Sensoren können eine unterschiedliche Anzahl von Werten messen, in den Experimenten kommen ein- und zweidimensionale Werte vor. Damit diese in einem geschlossenen Modell behandelt werden können, wird jede Komponente einzeln betrachtet. Da die Modelle datengetrieben sind, ist es nicht notwendig, statistische Unabhängigkeit zwischen den Komponenten anzunehmen.

Die Sensoren nehmen die Daten physikalisch unabhängig voneinander auf, das heißt, sie senden nicht in vordefinierter Reihenfolge, sondern dann, wenn ein Messwert vorliegt oder ein fest vorgegebener Zeitpunkt erreicht wird, unabhängig davon, ob andere Sensoren in diesem Netzwerk ihren Wert bereits gesendet haben. Daraus resultiert, dass die Messungen nicht synchron und nicht vollständig sind. Damit ein reguläres zeitdiskretes Signal verwendet werden kann, werden Zeitintervalle als Signalquellen betrachtet. Dafür werden jede Minute die Werte einer vergangenen Stunde gemittelt. Das bedeutet, es werden zu jedem Zeitpunkt n die Messwerte gemittelt, deren Zeitindex größer als $n - 60$ und kleiner als n ist, wobei der zeitliche Abstand $|t_n - t_{n-1}|$ eine Minute beträgt. Das Er-

gebnis ist für jeden Sensor ein Messwert pro Minute. Dadurch wird ein interpretierbares Signal erzeugt.

Dieses Signal wird quantisiert, wie in (5.6) beschrieben. Durch dieses Verfahren wird auch eine eventuelle ungleichmäßige Skalierung der Messwerte aus den Messungen entfernt. Für jeden Messwert wird ein Vektor mit Merkmalen erzeugt, die in einem CRF verwendet werden können. Anschließend werden diese Vektoren in einen kombinierten Merkmalsvektor zusammengefasst.

Das Intervall, über das die Messwerte gemittelt werden, beträgt 60 Minuten. Da die Sensoren mindestens einen Messwert pro Minute aufnehmen, werden in einem Intervall 60 Messwerte erwartet. Werden weniger als 80% dieser Werte erreicht, wird angenommen, dass der Sensor nicht zuverlässig arbeitet. Deswegen werden die zu diesem Sensor gehörigen Merkmale auf 0 gesetzt, also nicht in der Klassifikation berücksichtigt.

6.4.3 Modellbeschreibung, Training und Interpretation

Das Modell ist ein CRF mit $K + 1$ Färbungen, also K Färbungen für den Normalfall, eine weitere enthält Beschreibungen für das Ereignis. Dieses Modell wurde bereits in Abschnitt 6.1 besprochen. Das Training erfolgt überwacht, also nicht mittels des vorgestellten blinden Verfahrens. Das ist vor allem damit begründet, dass die Beobachtungen Verhaltensmustern von Personen entsprechen und damit die Zustände mit bestimmtem Verhalten zusammenhängen. Angenommen wird, dass Menschen zu festen Zeitpunkten denselben Gewohnheiten nachgehen, in etwa, dass sie des Nachts schlafen oder gegen Mittag eine Mahlzeit zu sich nehmen. Auch wenn der Rhythmus und die Gewohnheiten selbst sich zwischen den Personen unterscheiden können, so sind die Zeitpunkte dieser Aktionen in gewissen Intervallen konstant. Dabei ist es ferner möglich, dass die Person nicht jeden Tag diese Aktionen durchführt, wie zum Beispiel, dass sie an einigen Tagen die Wohnung zu bestimmten Zeitpunkten verlässt, an anderen nicht. Die Messwerte sind demnach an diesen Tagen unterschiedlich, solche können in vielen Verfahren nicht in einer einzigen Färbung berücksichtigt werden. CRFs, als deskriptive Modelle, die auf testbaren Informationen anstelle von metrischen Räumen basieren, können diese Diversität jedoch berücksichtigen. Hierbei wird getestet, ob entweder die eine oder die andere Aktion zu diesem Zeitpunkt eingetreten ist, wobei beide Aktionen zum Normalfall gehören.

Neben besonderen Abweichungen in den Messwerten wird die Gewohnheit zur Ereignisdetektion verwendet. Hat die Person eine Aktion zu einem bestimmten Zeitpunkt durchgeführt, und wird diese Aktion jedoch zu einem anderen detektiert, so wurde eine

starke Abweichung von den Gewohnheiten beobachtet; also ist das Ereignis auch vom Zeitpunkt abhängig, nicht nur von der Ausprägung selbst.

Um diese Zeitabhängigkeit zu berücksichtigen, werden die Zustände bestimmten Zeitintervallen zugeordnet. Es werden $K = 12$ Intervalle mit gleicher Länge verwendet. Dadurch kann eine Trainingssequenz aus den Zeitpunkten der Merkmale abgeleitet werden.

Dieses hat ebenfalls Einfluss auf die Interpretation der Messwerte. Da zu jedem Zeitpunkt der Zustand, an dem eine Beobachtung stattfindet, bekannt ist, kann die Interpretation in dem Sinne vereinfacht werden, dass nur dieser erwartete Zustand ausgewertet wird. Sinkt dessen Likelihood unter einen bestimmten Wert, so wird ein Ereignis angenommen. Als Messwert wird hier

$$v(n, \zeta_k) = \log\left(P(s(n) = \zeta_k|\mathbf{X}) \cdot (K + 1)\right) \tag{6.19}$$

verwendet; also die Likelihood, dass der Zustand zum Zeitpunkt n den Wert ζ_k annimmt, wobei für ζ_k immer die zum aktuellen Zeitpunkt gehörige Färbung verwendet wird. Diese Likelihood wird um $K + 1$ skaliert, um ein von der Anzahl der Zustände unabhängiges Maß zu erhalten, anschließend logarithmiert, um die Interpretation zu erleichtern: Ein negativer Wert weist auf das Ereignis hin, ein positiver Wert auf den Normalfall. Allerdings lassen sich die Aktionen unterschiedlich gut messen, daher sind Schwankungen in diesem Modell möglich, weswegen noch kein Schwellwert zur Interpretation vorgestellt wird. Es handelt sich hierbei um eine Machbarkeitsstudie, in der der Effekt der zeitlichen Abhängigkeit ebenfalls gemessen werden soll. Eine abschließende Betrachtung war zu dem gegebenen Zeitpunkt aufgrund des frühen Stadiums der Sensoren nicht möglich.

6.4.4 Ergebnisse

Das Experiment umfasst zwei Teile. Es wird auf Messwerten durchgeführt, die innerhalb von drei Monaten aufgenommen wurden, wobei die Messwerte des ersten Monats als Trainingsdaten verwendet werden.

Im ersten Teil handelt es sich um einen Test, ob das Modell prinzipiell zur Ereignisdetektion geeignet ist. Dafür wird ein besonders verlässlicher Datensatz genommen, also ein Datensatz aus einem Haushalt mit möglichst geringen Ausfällen der Messungen. Das Modell wurde auf Aufnahmen mehrerer Tage trainiert. Anschließend wurden die Messwerte manipuliert, um ein Ereignis zu simulieren. Dazu wurden auf die Sensorwerte ein normalverteiltes Rauschen addiert, dessen Varianz die doppelte Varianz der

Messwerte ist. Dieses Rauschen entspricht allerdings nicht realistischen Ereignissen: Aufgrund der Integration über große Zeitintervalle ist dieses Rauschen ein eher schlecht messbares Ereignis. Andere Ereignisse, wie nicht zuvor gemessene Werte, können als einfacher zu detektieren angenommen werden. Reale Ereignisse sind in der Menge der aufgenommenen Daten nicht vorhanden. Es ist somit nur ein prinzipieller und explizit schwierig gestellter Test, um die Aussagekraft dieses Experiments so hoch wie möglich zu halten. Ein Erfolg dieses Teils des Experiments ist gegeben, wenn das Maß $v(n, \zeta_k)$ deutlich sinkt.

Der zweite Teil behandelt den möglichen Ausfall von Sensoren. Hierbei werden Messwerte eines Sensors aus dem Datensatz entfernt, um den Ausfall zu simulieren. Dieses ist ein realistisches Szenario und daher der wichtigere Teil des Experiments. Ein Erfolg dieses Teils des Experiments ist gegeben, wenn das Maß $v(n, \zeta_k)$ zu dem gegebenen Intervallen möglichst wenig sinkt. Hieraus folgt, dass das Verfahren in Fällen von fehlender Information zur Ereignisdetektion eingesetzt werden kann. Dieses ist ein sehr wichtiger Aspekt für die Probleme der Ereignisdetektion, da, wie dieses Experiment zeigt, es ein realistisches Szenario ist, dass nicht immer alle Informationen gemessen werden.

Das Ergebnis des ersten Teils ist in Abbildung 6.6 zu sehen. Das Ereignis wurde in dem markierten Bereich simuliert. Nicht dargestellt ist der Bereich, auf dem trainiert wurde; es handelt sich also ausschließlich um eine Interpretation neuer Daten. Die deutlichen Änderungen der Messungen nach einem bestimmten Zeitpunkt sind einer Änderung der Systemkonfiguration geschuldet. Da es sich um eine Machbarkeitsstudie handelt, ist auch dieser Aspekt interessant: Hier wird deutlich die Sensitivität des Modells auf Änderungen gezeigt. Diese ist wichtig, um Ereignisse, die zum Teil schlecht messbar sind, zu erkennen. Die Ergebnisse sind daher als vielversprechend zu interpretieren. Genauere Interpretationen werden nur langfristige Studien liefern können, bei denen die Sensoren über Monate und eventuell Jahre eingesetzt werden und die im Rahmen dieser Arbeit daher nicht möglich sind.

Die deutliche Änderung des vorgestellten Maßes zeigt, dass das Modell prinzipiell geeignet ist, um Ereignisse in diesem Szenario zu erkennen. Eine Auswahl aus diesen Experimenten befindet sich in Abbildung 6.7 und 6.8. Die geringe Veränderung bei unterschiedlichen Sensoren zeigt, dass das Verfahren auch geeignet ist, falls Sensoren ausfallen, das heißt, im Fall von fehlenden Informationen.

Beide Teile des Experiments verliefen erwartungsgemäß. Daher kann die Einsatzmöglichkeit der Verfahren in diesen praktischen Problem angenommen werden.

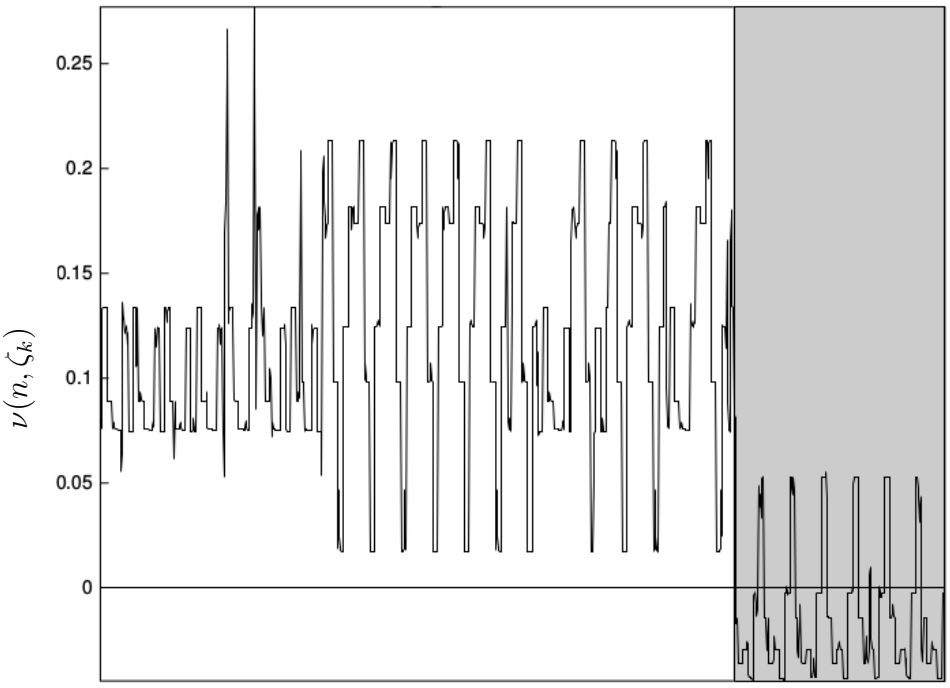


Abbildung 6.6: Ergebnisse des simulierten Ereignisses. Der markierte Bereich entspricht dem Zeitraum, in dem das Ereignis simuliert wurde. Der deutliche Abfall des präsentierten Maßes zeigt, dass das Verfahren prinzipiell für die Ereignisdetektion in diesem Szenario geeignet ist.

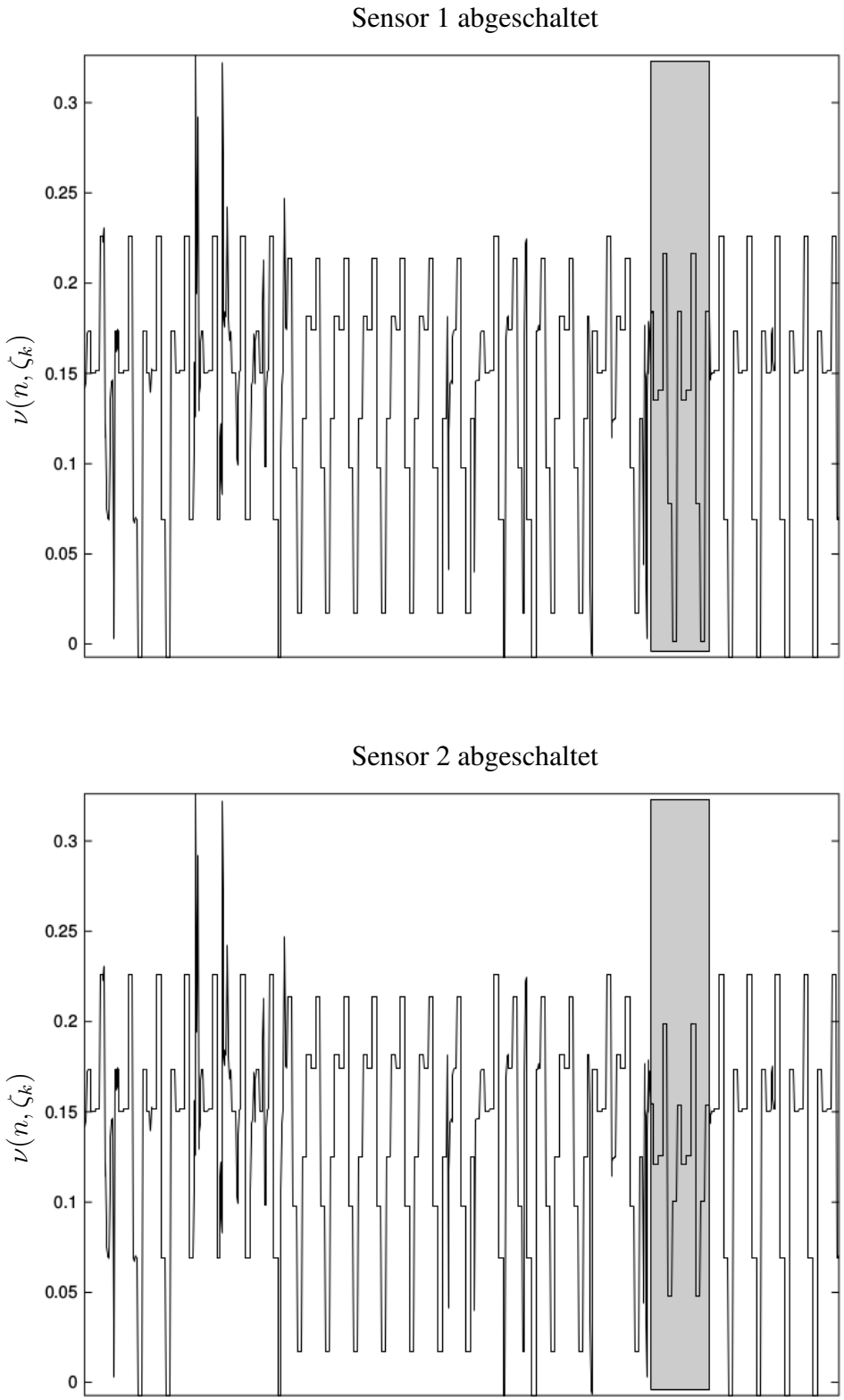
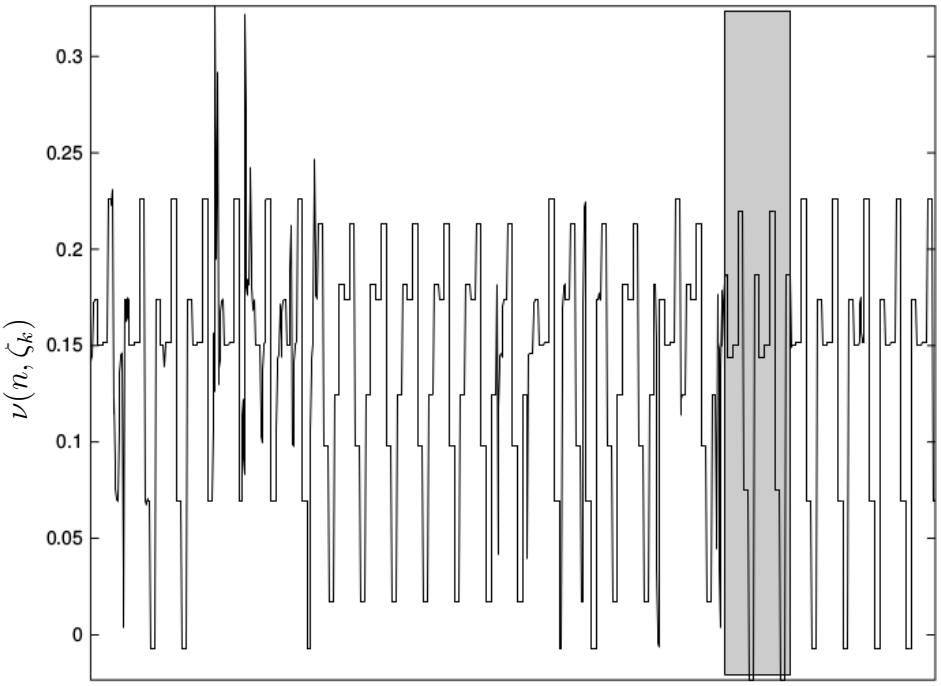


Abbildung 6.7: Ergebnis mit einem entfernten Sensor im markierten Bereich (1).

Sensor 3 abgeschaltet



Sensor 4 abgeschaltet

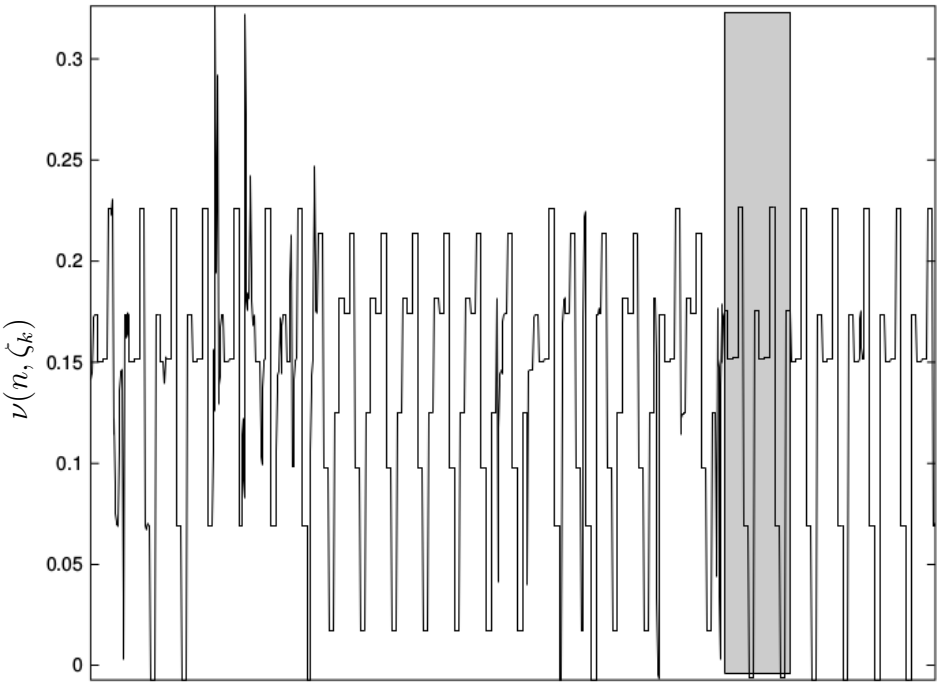


Abbildung 6.8: Ergebnis mit einem entfernten Sensor im markierten Bereich (2).

7 Zusammenfassung und Interpretation der Methoden und Experimente

In dieser Arbeit wurden zwei unterschiedliche Konzepte zur Ereignisdetektion behandelt. Beide Konzepte sind neu entwickelte Methoden, die speziell für die Detektion von unbekannten Ereignissen konzipiert wurden. Sie umfassen Algorithmen zur Behandlung von stetigen Signalen und diskreten Werten bis hin zu sehr unterschiedlichen Mischverteilungen und decken damit einen großen Teil aller möglichen Signale ab.

Das erste Verfahren basiert auf dem Prinzip der linearen Vorhersage. Dabei wird ein ein- oder mehrdimensionales Signal stückweise analysiert, indem Abweichungen zu erwarteten Messwerten interpretiert werden. Mit einem Satz von Prädiktoren wird ein weiterer Messwert vorhergesagt, und die Abweichung zum neuen tatsächlichen Wert gibt Auskunft über die Möglichkeit eines Ereignisses. Die Prädiktoren werden dabei in ein Maß zusammenfasst, das als Maß für Normalität interpretiert werden kann.

Dieses Verfahren setzt eine stückweise Stationarität des Signals voraus sowie eine lineare Abhängigkeit in kurzen Abschnitten. Für viele Probleme mit praktisch relevanten Abtastraten ist diese Annahme gegeben. Dadurch ist diese Methode durchaus in der Praxis anwendbar, etablierte Verfahren wie GMMs und HMMs stellen zum Teil deutlich restriktivere Bedingungen.

Es kommen Problemstellungen vor, bei denen die Annahmen letzterer Verfahren nicht haltbar sind, in etwa, wenn kategorische Messwerte (zum Beispiel “Tür ist auf”) oder Zählgrößen (“Tür wurde im letzten Zeitintervall n mal geöffnet”) verwendet werden. Diese lassen sich nicht mit solchen Modellen behandeln, die eine stückweise Stationarität voraussetzen. Sie sind andererseits von praktischer Relevanz, weswegen andere Methoden nützlich sind. Ein Ansatz hierfür wurde in dieser Arbeit entwickelt, das sind die CRF-basierten Methoden. Diese können deutlich umfassender auf beliebige Daten angewendet werden. Die CRF-basierten Methoden transformieren die Daten in Wahrscheinlichkeiten der Zustände. Dadurch wird die Interpretation der Information

einfacher. Diese Methode erlaubt es also, sehr komplexe Zusammenhänge geschlossen zu analysieren und bietet eine einfache Plattform für viele Erweiterungen.

Letztere Verfahren sind graphische Methoden. Daraus folgt die einfache Interpretation: Die Zustände nehmen Farben an, wobei im einfachsten Fall eine der Farben ausdrückt, dass das Ereignis eingetreten ist. Andere Möglichkeiten, Ereignisse zu bestimmen, sind unwahrscheinliche Abfolgen, zum Beispiel, dass ein Segment von Zuständen, die alle dieselbe Farbe angenommen haben, zu lang oder zu kurz ist. Dadurch können sehr unterschiedliche Ereignisse erfasst werden, und unterschiedliche Interpretationen können je nach Anwendungsfall sehr einfach entwickelt werden.

Einer der wichtigsten Beiträge dieser Arbeit ist, dass das Training der CRFs für den Anwendungsfall der Ereignisdetektion modifiziert worden ist. Hierfür wurden zwei unterschiedliche Methoden vorgestellt. Eine Möglichkeit ist es, dass der Strafterm für die logarithmierte Likelihood modifiziert wird, indem eine Strafmatrix hinzugenommen wird. Dieses ist eine sehr anpassbare Methode, bedingt jedoch höheren Aufwand bei der Optimierung. Die andere Möglichkeit basiert auf einer restringierten Optimierung. Diese ist einfacher und für die meisten Probleme geeignet, allerdings bietet sie weniger Anpassungsmöglichkeiten. Insgesamt lassen sich sehr unterschiedliche Fälle behandeln, daher können Algorithmen, die auf diesem Modell beruhen, sehr schnell entwickelt werden.

Insbesondere wurde auch ein Verfahren zum unüberwachten Training vorgestellt. Dieser Ansatz geht einher mit dem Prinzip der maximalen Entropie, die auch die Basis für die hier vorgestellten Modelle bildet [54]. Es wurde explizit vermieden, unnötige Annahmen in das System aufzunehmen, was eine Verallgemeinerung der Verfahren ermöglicht. Dadurch können die Methoden ohne besondere Vorsicht für weitere Problemstellungen übernommen werden.

In den Experimenten zeigten sich die Methoden als für ihr jeweiliges Einsatzgebiet geeignet. Das Mischmodell der linearen Prädiktoren ist einfach anzuwenden und bietet die Möglichkeit, zeitliche Abfolgen von Messwerten zu analysieren [52]. Die Markov-Modelle eignen sich für diverse Überwachungsprobleme [48, 49, 51], zudem gibt es die Möglichkeit, mit Hilfe der Modelle Sequenzen in Cluster zu bündeln [50]. Dadurch bieten die vorgestellten Methoden ein breites Spektrum an möglichen Anwendungen.

8 Ausblick

Die vorgestellten Methoden sind dafür konzipiert, eine Basis für viele unterschiedliche Probleme zu bilden. Die Experimente sind daher nur beispielhaft für viele gewählt, die Anwendungen können nicht erschöpfend diskutiert werden. Das Spektrum der Anwendungsfälle, die hier als Beispiel dienten, umfasst vor allem Sensornetze und Videosequenzen, da diese Felder insbesondere für die Überwachungssysteme wichtig sind.

Am Rand wurden ebenfalls Audiosignale behandelt. Es wurde gezeigt, dass das Clusterverfahren für diese Signale geeignet ist; hiernach erhält man eine Zustandssequenz, das ist eine Abstraktionsebene, die ebenfalls im Sinne der Ereignisdetektion interpretiert werden kann. Daher ist die Behandlung von Audioereignissen mittels dieser Methoden möglich. Da Audiosignale neben Videosignalen wichtige Beispiele für die Ereignisdetektion sind, ist dieses Ergebnis positiv zu bewerten. Für einen konkreten Anwendungsfall kann daher eine auf CRFs basierende Methode, wie sie in dieser Arbeit diskutiert wurde, verwendet werden.

Die meisten hier vorgestellten Methoden basieren darauf, komplexe Daten in eine abstrakte Ebene zu transformieren, die anschließend mit gemeinsamen, einfacheren Methoden analysiert werden können. Diese Abstraktion ist anwendungsabhängig und muss für die Probleme angepasst werden. Das ist ein Problem der Merkmalsextraktion, dieser Punkt ist somit für alle Methoden der Ereignisdetektion zu beachten. Anschließend können diverse Problemstellungen äquivalent der in dieser Arbeit diskutierten Interpretationsmöglichkeiten behandelt werden. Die hier vorgestellten Verfahren sind demnach sehr anpassungsfähig, da sie nur noch Probleme der Merkmalsextraktion für das konkrete Problem offen lassen, um vollständige Systeme der Ereignisdetektion zu bieten.

Neben der Ereignisdetektion sind viele der hier vorgestellten Methoden auch für andere Entwicklungen interessant. Eine Anwendung abseits dieser bietet vor allem das unüberwachte Training der CRFs. Eine häufige Modifikation von CRFs sind vielschichtige CRFs, das sind solche Modelle, in denen mehrere Ebenen von Markov-Ketten in einem Modell zusammengefasst werden [64, 81]. Als Eingabe dient hierbei eine

Sequenz von Messwerten, des Weiteren wird für das Training eine bekannte Färbung einer einzelnen Ebene angenommen. Hierbei werden oft GMMs verwendet, um die Ebenen zwischen der Eingabeschicht und der Ausgabeschicht, deren Färbung bekannt ist, zu färben [75]; anschließend werden die Parameter der Zwischenschicht separat trainiert. Andere Verfahren übernehmen dieselbe Färbung für die Zwischenschichten [86] oder verwenden aufwendige Methoden, die nicht schnell konvergieren [81]. Für diese bietet das unüberwachte Training eine effiziente Alternative. Dadurch erweitert sich das Spektrum der Anwendungsmöglichkeiten der CRFs beträchtlich.

Das erste Verfahren, die lineare Vorhersage, steht zunächst separat dar. Allerdings zeigt sich, dass die einzelnen Testfunktionen Werte im Intervall zwischen 0 und 1 annehmen. Das ist ebenfalls das Intervall, das für die Merkmalsextraktion für CRFs genutzt wird.

In diesem Sinne sind auch die Kombinationen der Verfahren, die hier vorgestellt wurden, nur exemplarisch zu sehen. Die Sequenzen lassen sich mit unterschiedlicher Länge analysieren, die Merkmalsextraktion an auftretende Probleme anpassen; dadurch sind echtzeitfähige und auch nachträgliche Interpretationen möglich. Diese Vielseitigkeit, die sich nicht durch eine begrenzte Anzahl an Experimenten erläutern lässt, ist das wichtigste Argument für die hier vorgestellten Methoden und bieten die Möglichkeit, eine Basis neuer Verfahren zur Ereignisdetektion zu sein.

Literatur

[1] Norm Aleks u. a. *Probabilistic detection of short events, with application to critical care monitoring*. Hrsg. von D. Koller u. a. 2009.

[2] Yasemin Altun, Ioannis Tsochantaridis und Thomas Hofman. *Hidden Markov Support Vector Machines*. 2003.

[3] Jeffrey D. Banfield und Adrian E. Raftery. „Model-Based Gaussian and Non-Gaussian Clustering“. In: *Biometrics* 49.3 (1993), S. 803–821.

[4] M. Basseeville und Igor V. Nikiforov. „Detection of Abrupt Changes: Theory and Application“. In: Prentice-Hall, 1993, 35ff.

[5] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006. ISBN: 0387310738.

[6] Hans L. Bodlaender. „On the Complexity of Some Coloring Games.“ In: *WG*. Hrsg. von Rolf H. Möhring. Bd. 484. Lecture Notes in Computer Science. Springer, 1990, S. 30–40. ISBN: 3-540-53832-1.

[7] O. Boiman und M. Irani. „Detectiong irreguarities in images and in video“. In: *IJCV*. 2007, S. 17–31.

[8] Stephen Boyd und Lieven Vandenberghe. *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004. ISBN: 0521833787.

[9] Ivana M. Burazor und Mirko Burazor. „Troponin T in Unstable Angina Pectoris“. In: *Facta Universitatis*. Bd. 9. Medicine and Biology 3. 2002, S. 240–244.

[10] Y. Censor und University of Pennsylvania. Department of Radiology. *On Iterative Methods for Linearly Constrained Entropy Maximization*. Medical Image Processing Group technical report. Department of Radiology, University of Pennsylvania, 1986. URL: `http://books.google.de/books?id=t-AmHQAACAAJ`.

[11] Jackie Chi Kit Cheung und Xiao Li. „Sequence clustering and labeling for unsupervised query intent discovery“. In: *Proceedings of the fifth ACM international conference on Web search and data mining*. WSDM '12. Seattle, Washington, USA: ACM, 2012, S. 383–392. ISBN: 978-1-4503-0747-5. DOI: 10.1145/2124295.2124342.

[12] I Cohen und G Medioni. „Detecting and tracking moving objects for video surveillance“. In: *Proceedings 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Cat No PR00149* 2 (1999), S. 319–325.

[13] A. Condurache u. a. „Fast Detection and Processing of Arbitrary Contrast Agent Injections in Coronary Angiography and Fluoroscopy“. In: *Bildverarbeitung für die Medizin (Algorithmen, Systeme, Anwendungen)*. 2004, S. 5–9.

[14] A. P. Condurache. *Cardiovascular Biomedical Image Analysis: Methods and Applications*. ISBN 978-3-89863-236-2. Waabs, Germany: GCA-Verlag, Aug. 2008.

[15] Alexandru Paul Condurache und Alfred Mertins. „A point-event detection algorithm for the analysis of contrast bolus in fluoroscopic images of the coronary arteries“. In: *Proc. EUSIPCO 2009*. Glasgow, 2009, S. 2337–2341.

[16] Nello Cristianini und John Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. 1. Aufl. Cambridge University Press, 2000. ISBN: 0521780195.

[17] Arnaud Doucet u. a. „Rao-Blackwellised Particle Filtering for Dynamic Bayesian Networks“. In: *UAI '00: Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2000, S. 176–183. ISBN: 1-55860-709-9.

[18] Tommy Elfving. „On some methods for entropy maximization and matrix scaling“. In: *Linear Algebra and its Applications* 34.0 (1980), S. 321 –339.

[19] E.R.Dougherty. *Math Morphology in Image Processing*. New York: Marcel Dekker, 1992.

[20] Maurizio Filippone und Guido Sanguinetti. *Novelty Detection in Autoregressive Models using Information Theoretic Measures*. Techn. Ber. Department of Computer Science, University of Sheffield, 2009.

[21] R. Fletcher. *Practical methods of optimization; (2nd ed.)* New York, NY, USA: Wiley-Interscience, 1987. ISBN: 0-471-91547-5.

[22] A. Freno und E. Trentin. *Hybrid Random Fields: A Scalable Approach to Structure and Parameter Learning In Probabilistic Graphical Models*. Intelligent Systems Reference Library. Springer Berlin Heidelberg, 2011. ISBN: 9783642203084.

[23] William Gibbs. *Elementary Principles in Statistical Mechanics*. Hrsg. von Camebridge University Press. Camebridge University Press, 1928.

[24] Yves Grandvalet und Yoshua Bengio. *Semi-supervised Learning by Entropy Minimization*. Cambridge, MA, 2005.

[25] Yu Gu, Andrew McCallum und Don Towsley. „Detecting anomalies in network traffic using maximum entropy estimation“. In: *Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement*. IMC '05. Berkeley, CA, USA: USENIX Association, 2005, S. 32–32.

[26] J. Gupchup u. a. *Model-Based Event Detection in Wireless Sensor Networks*. Jan. 2009. arXiv: 0901.3923 [cs.NI].

[27] Kapil Kumar Gupta, Baikunth Nath und Kotagiri Ramamohanarao. „Conditional Random Fields for Intrusion Detection“. In: *Proceedings of 21st International Conference on Advanced Information Networking and Applications Workshops (AINAW)*. IEEE Press, 2007, S. 203–208.

[28] Rahul Gupta. *Conditional Random Fields*. Technical report, IIT Bombay. 2006.

[29] Fredrik Gustafsson. *Adaptive Filtering and Change Detection*. John Wiley und Sons, Inc., 2000.

[30] D Gutchess u. a. „A Background Model Initialization Algorithm for Video Surveillance“. In: *Proceedings Eighth IEEE International Conference on Computer Vision ICCV 2001* 00.C (2001), S. 733–740.

[31] G. Heigold u. a. „Discriminative HMMs, Log-Linear Models, and CRFs: What is the Difference?“ In: *IEEE International Conference on Acoustics, Speech, and Signal Processing*. Dallas, Texas, USA, März 2010, S. 5546–5549.

[32] S. Ikbal und T. Faruquie. „HMM based event detection in audio conversation“. In: *IEEE International Conference on Multimedia and Expo* (2008), S. 1497–1500.

[33] Yuri Ivanov u. a. „Video Surveillance of Interactions“. In: *IN CVPR WORKSHOP ON VISUAL SURVEILLANCE, FORT COLLINS*. IEEE, 1998, S. 82–89.

[34] Kenneth E. Iverson. „A Programming Language“. In: John Wiley & Sons, Inc., 1962, S. 11.

[35] E. T. Jaynes. „Information Theory and Statistical Mechanics - Jaynes“. In: *The Physical Review* 106.4 (1957), S. 620–630.

[36] JamesM. Joyce. „Kullback-Leibler Divergence“. English. In: *International Encyclopedia of Statistical Science*. Hrsg. von Miodrag Lovric. Springer Berlin Heidelberg, 2014, S. 720–722. ISBN: 978-3-642-04897-5. DOI: 10.1007/978-3-642-04898-2_327.

[37] A. Juneja und Espy C. Wilson. „Segmentation of continuous speech using acoustic-phonetic parameters and statistical learning“. In: *Proceedings of the 9th International Conference on Neural Information Processing, 2002. ICONIP '02.* Bd. 2.

[38] Imran N. Junejo, Omar Javed und Mubarak Shah. „Multi Feature Path Modeling for Video Surveillance“. In: *Pattern Recognition, International Conference on* 2 (2004), S. 716–719. ISSN: 1051-4651. DOI: 10.1109/ICPR.2004.1334359.

[39] R E Kalman. „A new approach to linear filtering and prediction problems“. In: *Journal Of Basic Engineering* 82.Series D (1960). Hrsg. von H WEditor Sorenson, S. 35–45.

[40] J. Kwon und K. M. Lee. „Simultaneous video synchronization and rare event detection via Cross-Entropy Monte Carlo optimization“. In: *VS*. 2009, S. 1322–1329.

[41] J. D. Lafferty, A. McCallum und F. C. N. Pereira. „Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data“. In: *Proceedings of the Eighteenth International Conference on Machine Learning*. ICML '01. San Francisco, CA, USA, 2001, S. 282–289. ISBN: 1-55860-778-1.

[42] Mo Li, Yunhao Liu und Lei Chen. „Non-Threshold based Event Detection for 3D Environment Monitoring in Sensor Networks“. In: *Proceedings of the 27th International Conference on Distributed Computing Systems*. ICDCS '07. Washington, DC, USA: IEEE Computer Society, 2007, S. 9–. ISBN: 0-7695-2837-3. DOI: 10.1109/ICDCS.2007.123.

[43] Wanrong Liu und Xuewen Lu. „Weighted least squares method for censored linear models“. In: *Journal on Nonparametric Statistics*. Bd. 21. 2009, S. 787–799.

[44] Carla Lopes und Fernando Perdigão. „Event Detection by HMM, SVM and ANN: A Comparative Study“. In: *Proceedings of the 8th international conference on Computational Processing of the Portuguese Language*. PROPOR '08. Berlin, Heidelberg: Springer-Verlag, 2008, S. 1–10. ISBN: 978-3-540-85979-6. DOI: 10.1007/978-3-540-85980-2_1.

[45] Robert Malouf. „A Comparison of Algorithms for Maximum Entropy Parameter Estimation“. In: *Proceedings of the 6th Conference on Natural Language Learning - Volume 20*. COLING-02. Stroudsburg, PA, USA: Association for Computational Linguistics, 2002, S. 1–7.

[46] *Switching kalman filters for prediction and tracking in an adaptive meteorological sensing network*. 2005, S. 197–206. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1557075.

[47] Markos Markou und Sameer Singh. „Novelty Detection: A Review - Part 1: Statistical Approaches“. In: *Signal Processing* 83 (2003), S. 2003.

[48] D. Matern, A. P. Condurache und A. Mertins. „Adaptive and Automated Ambiance Surveillance and Event Detection for Ambient Assisted Living“. In: *Proc. Conf. on Eng. in Med. and Biol. Soc. (EMBC)*. Osaka, Japan, 2013, S. 7318–7321.

[49] D. Matern, A. P. Condurache und A. Mertins. „Automated Intrusion Detection for Video Surveillance Using Conditional Random Fields“. In: *Proc. Conf. on Mach. Vis. Appl. (MVA)*. Kyoto, Japan, 2013, S. 298–301.

[50] D. Matern, A. P. Condurache und A. Mertins. „Automated Sequence Clustering of Audio Signals using Conditional Random Fields“. In: *Proc. AIA-DAGA 2013 Conference on Acoustics*. Merano, Italy, 2013, S. 1439–1440.

[51] D. Matern, A. P. Condurache und A. Mertins. „Event Detection using Log-Linear Models for Coronary Contrast Agent Injections“. In: *Proceedings of the First International Conference on Pattern Recognition Applications and Methods (ICPRAM)*. Bd. 2. Vilamoura - Algarve, Portugal, 2012, S. 172–179.

[52] D. Matern, A. P. Condurache und A. Mertins. „Linear Prediction based Mixture Models for Event Detection in Video Sequences“. In: *Proc. Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA) 2011*. Gran Canaria, Spain, 2011, S. 25 –32.

[53] A. McCallum. „Efficiently inducing features of conditional random fields“. In: *Nineteenth Conference on Uncertainty in Artificial Intelligence (UAI03)*. 2003.

[54] A. McCallum, D. Freitag und F. C. N. Pereira. „Maximum Entropy Markov Models for Information Extraction and Segmentation“. In: *Proceedings of the Seventeenth International Conference on Machine Learning*. ICML '00. San Francisco, CA, USA, 2000, S. 591–598. ISBN: 1-55860-707-2.

[55] Ryan McDonald und Fernando Pereira. „Identifying gene and protein mentions in text using conditional random fields“. In: *BMC Bioinformatics* 6.Suppl 1 (2005), S6+. ISSN: 1471-2105.

[56] G. J. McLachlan und T. Krishnan. *The EM algorithm and extensions*. 2. Wiley series in probability and statistics. Hoboken, NJ: Wiley, 2008. XXVII, 359. ISBN: 978-0-471-20170-0.

[57] Alfred Mertins. *Signaltheorie*. 2. Auflage. Vieweg+Teubner, 2010.

[58] Vivek Mhatre und Catherine Rosenberg. „Homogeneous vs. heterogeneous clustered sensor networks: A comparative study“. In: *In Proceedings of 2004 IEEE International Conference on Communications (ICC 2004*. 2004, S. 3646–3651.

[59] Vinoid Nair und James J. Clark. *Automated Visual Surveillance Using Hidden Markov Models*. Techn. Ber. Centre for Intelligent Machines, McGill University Montreal, 2000.

[60] Maren Pakura, Oliver Schmitt und Til Aach. „Segmentation and Analysis of Nerve Fibers in Histologic Sections of the Cerebral Human Cortex.“ In: *SSIAI*. 2002, S. 62–66.

[61] Fuchun Peng, Fangfang Feng und Andrew McCallum. *Chinese Segmentation and New Word Detection Using Conditional Random Fields*. 2004.

[62] Massimo Piccardi. „Background subtraction techniques: a review“. In: *SMC (4)*. 2004, S. 3099–3104.

[63] Stephen Della Pietra, Vincent Della Pietra und John Lafferty. „Inducing Features of Random Fields“. In: *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE* 19.4 (1997), S. 380–393.

[64] Ariadna Quattoni, Michael Collins und Trevor Darrell. „Conditional random fields for object recognition“. In: *In NIPS*. MIT Press, 2004, S. 1097–1104.

[65] L. Rabiner. „A tutorial on hidden Markov models and selected applications in speech recognition“. In: Bd. 77. 2. 1989, S. 257–286. DOI: 10.1109/5.18626.

[66] Alvin C. Rancher. *Linear Models in Statistics*. John Wiley und Sons, Inc., 2000.

[67] Seyed Saeed Changiz Rezaei. „Entropy and Graphs“. Magisterarb. University of Waterloo, 2013.

[68] Wang Shitong u. a. „Robust maximum entropy clustering algorithm with its labeling for outliers“. In: *Soft Comput.* 10.7 (Mai 2006), S. 555–563. ISSN: 1432-7643. DOI: 10.1007/s00500-005-0517-5.

[69] Noah A. Smith. *Log-Linear Models*. 2004.

[70] Ying-Li Tian, Max Lu und Arun Hampapur. „Robust and Efficient Foreground Analysis for Real-Time Video Surveillance“. In: *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*. CVPR '05. Washington, DC, USA: IEEE Computer Society, 2005, S. 1182–1187. ISBN: 0-7695-2372-2.

[71] Vladimir N. Vapnik. *The nature of statistical learning theory*. New York, NY, USA: Springer-Verlag New York, Inc., 1995. ISBN: 0-387-94559-8.

[72] Régis Vert und Jean-Philippe Vert. „Consistency and Convergence Rates of One-Class SVMs and Related Algorithms“. In: *J. Mach. Learn. Res.* 7 (Dez. 2006), S. 817–854. ISSN: 1532-4435.

[73] Gilbert Walker. „On Periodicity in Series of Related Terms“. In: *Proceedings of the Royal Society of London*. Bd. 131. 1931, S. 518–532.

[74] Hanna M. Wallach. *Conditional random fields: An introduction*. Techn. Ber. University of Pennsylvania, 2004.

[75] Sy Bor Wang u. a. „Hidden Conditional Random Fields for Gesture Recognition“. In: *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2*. CVPR '06. Washington, DC, USA: IEEE Computer Society, 2006, S. 1521–1527. ISBN: 0-7695-2597-0.

[76] Greg Welch und Gary Bishop. *An Introduction to the Kalman Filter*. Techn. Ber. Chapel Hill, NC, USA: Department of Computer Science, University of North Carolina at Chapel Hill, 1995.

[77] D. B. West. *Introduction to Graph Theory (2nd Edition)*. Hrsg. von Prentice Hall. Prentice Hall, 2001.

[78] Christopher K. I. Williams, John A. Quinn und Neil Mcintosh. „Factorial Switching Kalman Filters for Condition Monitoring in Neonatal Intensive Care“. In: *Advances in Neural Information Processing Systems 18, NIPS 2005*. Vancouver, British Columbia, Canada, Dez. 2005.

[79] T. Xiang und S. Gong. „Video behavior prolling for anomaly detection“. In: *PAMI*. 2008, S. 893–908.

[80] J. Yang und W. Wang. „CLUSEQ: Efficient and Effective Sequence Clustering“. In: *Proceedings of the 19th International Conference on Data Engineering*. IEEE Press, 2003, S. 101–112.

[81] Dong Yu, Li Deng und Shizhen Wang. „Learning in the deepstructured conditional random fields“. In: *NIPS 2009 Workshop on Deep Learning for Speech Recognition and Related Applications*. 2009.

[82] G. Udny Yule. „On a Method of Investigating Periodicities in Disturbed Series, with Special Reference to Wolfer's Sunspot Numbers“. In: *Philosophical Transactions of the Royal Society of London*. Bd. 226. 1927, S. 267–298.

[83] Dengsheng Zhang und Guojun Lu. „A Comparative Study on Shape Retrieval Using Fourier Descriptors with Different Shape Signatures“. In: *Victoria* 1.January (2001), 1–9.

[84] Tong Zhang u. a. „Fall Detection by Wearable Sensor and One-Class SVM Algorithm“. In: *Intelligent Computing in Signal Processing and Pattern Recognition*. Hrsg. von De-Shuang Huang, Kang Li und GeorgeWilliam Irwin. Bd. 345. Lecture Notes in Control and Information Sciences. Springer Berlin Heidelberg, 2006, S. 858–863. ISBN: 978-3-540-37257-8.

[85] Wei Zhang, Sen-Ching S. Cheung und Minghua Chen. „Hiding privacy information in video surveillance system.“ In: *ICIP (3)*. 2005, S. 868–871.

[86] Jun Zhao, Kang Liu und Gen Wang. „Adding redundant features for CRFs-based sentence sentiment classification“. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. EMNLP '08. Stroudsburg, PA, USA: Association for Computational Linguistics, 2008, S. 117–126.

[87] H. Zhong, M. Visontai und J. Shi. „Detecting unusual activity in video“. In: *CVPR*. 2004.

[88] Hanning Zhou und Don Kimber. „Unusual Event Detection via Multi-camera Video Mining“. In: *Proceedings of the 18th International Conference on Pattern Recognition - Volume 03*. ICPR '06. Washington, DC, USA: IEEE Computer Society, 2006, S. 1161–1166. ISBN: 0-7695-2521-0. DOI: 10.1109/ICPR.2006.1149.

[89] Xiaodan Zhuang u. a. „Acoustic fall detection using Gaussian mixture models and GMM supervectors“. In: *IEEE International Conference on Acoustics, Speech, and Signal Processing* 0 (2009), S. 69–72.

[90] Francesco Ziliani u. a. „Image Analysis for Video Surveillance Based on Spatial Regularization of a Statistical Model-Based Change Detection“. In: *in Proc. Int. Conf. on Image Analysis and Processing*. 1999, S. 1108–1111.