



Fakultät für Informatik
Institut für Simulation und Grafik
Lehrstuhl für Simulation

**Fallstudie zur Anwendung eines Hidden non-Markovian Model innerhalb einer
System Reliability Analysis**

Masterarbeit

Im Rahmen des Studiums der Computervisualistik

Bearbeitet von: Mathias Vorbröcker
An der Enckekaserne 8
39110 Magdeburg

Matrikelnummer: 178933
Zur Erlangung des akademischen Grades

Master of Engineering

Abgabedatum: 04.11.2013

ABSTRACT

Bereits einzelne Ausfälle von Systemkomponenten können zum Defekt ganzer Systeme und gesteigerten Instandhaltungskosten führen. Daher besteht ein Bedarf zur quantitativen Betrachtung der Zuverlässigkeit von Systemen, der zur Entstehung der System Reliability Analysis als eigenständige wissenschaftliche Disziplin führte. Diese vereint die Lehre der Wahrscheinlichkeit und der Statistik, um die Beständigkeit von Systemen und die Auswirkungen von Fehlern zu evaluieren. (vgl. Zio 2013, S. 7–8)

Bei partiell-beobachtbare diskrete stochastische Systeme (PODS) ist eine lückenlose Aufnahme des dynamischen Verhaltens nicht möglich oder unverhältnismäßig aufwendig. PODS können zwar mittels Hidden non-Markovian Models (HnMMs) dargestellt werden, deren Analyse ist jedoch zeitlich aufwendig, weswegen nach Methoden geforscht wurde, um PODS effizienter auszuwerten. Je nachdem, ob ein System bei sämtlichen Übergängen von einem Systemzustand in den anderen oder nur bei manchen Transitionen ein Output generiert wird, und ob der Einfluss der vergangenen Zeit auf die Distributionen der Transitionen nach bestimmten Ereignissen zurückgesetzt werden kann, wurden bereits verschiedene Lösungen erarbeitet, deren Tauglichkeit in der Praxis jedoch noch nicht erforscht wurde. (vgl. Buchholz 2012, S.1–2)

Daher wird im Rahmen dieser Arbeit eine Fallstudie durchgeführt, bei der eine CHnMM zur Analyse der Zuverlässigkeit eines bestimmten Systems verwendet wird. Zu diesem Zweck wird ein bestimmtes PODS durch Methoden der HnMMs ausgewertet, die am Lehrstuhl für Simulation der Otto-von-Guericke Universität entwickelt und veröffentlicht wurden.

Somit bietet diese Arbeit ein erstes Beispiel zur Umsetzung einer System Reliability Analysis mittels HnMMs und eine erste Einschätzung über die Angemessenheit der Anwendung einer solchen Analyse auf PODS. Die Arbeit stellt somit eine erste Erfahrungsgrundlage für die Durchführung von System Reliability Analysis dar, anhand derer Entscheidungsträger die CHnMM als Methode kennen lernen, die Eignung von vorliegenden Daten für diese Methode prüfen und den Aufwand der Durchführung sowie die Aussagekraft der Ergebnisse ihrer Studien einschätzen können.

INHALTSVERZEICHNIS

Abstract.....	I
Inhaltsverzeichnis	II
Abbildungsverzeichnis	IV
Tabellenverzeichnis	V
Formelverzeichnis	1
1 Einleitung	1
1.1 Ausgangslage	1
1.2 Zielsetzung.....	1
1.3 Aufbau der Arbeit	2
2 Grundlagen	3
2.1 System Reliability Analysis.....	3
2.1.1 Zuverlässigkeit technischer Systemen.....	3
2.1.2 Zuverlässigkeit einfacher Systeme	4
2.1.3 Zuverlässigkeit strukturierter Systeme	7
2.1.4 Die Verfügbarkeit instandsetzbarer Systeme	8
2.2 Hidden Markov Models	10
2.2.1 Discrete Time Markov Models.....	11
2.2.2 Spezifikation eines Hidden Markov Model.....	12
2.2.3 Anwendungsfall Evaluation	13
2.2.4 Anwendungsfall Dekodierung.....	14
2.2.5 Anwendungsfall Training.....	15
2.3 Hidden non-Markovian Models.....	18
2.3.1 Spezifikation eines Conversive Hidden non-Markovian Model	19
2.3.2 Anwendungsfall Evaluation	21
2.3.3 Anwendungsfall Decodierung	23
2.3.4 Anwendungsfall Training.....	23
3 Fallstudie einer System Reliability Analysis.....	26
3.1 Ausgangslage und Ziele der Studie.....	26
3.2 HMM der Produktionslinie	31
3.2.1 Struktur des Modells.....	31

3.2.2	Unsicherheit innerhalb der Symbolsequenzen	34
3.2.3	Implementation des HMM	36
3.3	Aufbau und Implementation der CHnMM	38
3.3.1	Struktur des Modells.....	38
3.3.2	Unsicherheit innerhalb der Symbolsequenzen	41
3.3.3	Implementation der CHnMM	43
3.4	Ergebnisanalyse	44
4	Rückblick und Ausblick	48
4.1	Zusammenfassung	48
4.2	Ausblick	49
	Literaturverzeichnis	VIII
	Selbstständigkeitserklärung	IX

ABBILDUNGSVERZEICHNIS

Abbildung 2.1 Zustandsänderung von einfachen Systemen ohne Instandsetzung.....	5
Abbildung 2.2 Beispielhafter Verlauf der Ausfallwahrscheinlichkeit $F(t)$	5
Abbildung 2.3 Beispielhafter Verlauf der Zuverlässigkeit $R(t)$ (vgl. Ryll 2011, S. 47)	6
Abbildung 2.4 Beispielhafter Verlauf der Ausfallrate $\lambda(t)$	7
Abbildung 2.5 Schaltbild eines Seriensystems.....	8
Abbildung 2.6 Zustandsgraph eines binären technischen Systems	9
Abbildung 3.1 DTMC einer Maschine von M	32
Abbildung 3.2 Beispiel zur Berechnung der Einträge des Outputmatrix	36
Abbildung 3.3 Zustandsraum der CH_nMM einer Maschine	39
Abbildung 3.4 Zustandsraum der CH_nMM der Arbeitsstation 7	40
Abbildung 3.5 Markovprozess der Outputmatrizen	42

TABELLENVERZEICHNIS

Tabelle 3.1 time between failure der Maschinen in 8-Stunden-Schichten	29
Tabelle 3.2 time to repair der Maschinen in Minuten	30
Tabelle 3.3 Trainingsergebnis HMM von S0 nach S4	45
Tabelle 3.4 CHnMM Training Ergebnisse	46

FORMELVERZEICHNIS

Formel 2.1 Zustand eines Systems	5
Formel 2.2 Verteilungsfunktion	5
Formel 2.3 Eigenschaften der Verteilungsfunktion.....	6
Formel 2.4 Ausfalldichte	6
Formel 2.5 Ausfallrate	7
Formel 2.6 Zuverlässigkeit eines Seriensystems.....	8
Formel 2.7 Ausfallrate eines Seriensystems.....	8
Formel 2.8 Verfügbarkeit eines Systems.....	9
Formel 2.9 Zustandsraum eines Zwei-Maschinen-Modells	10
Formel 2.10 Markov-Eigenschaft eines Modells	11
Formel 2.11 Zeithomogenität	11
Formel 2.12 Systemzustand.....	12
Formel 2.13 DTMC Schritt von t nach t +1	12
Formel 2.14 DTMC Schritt von 0 nach t.....	12
Formel 2.15 Outputmatrix	12
Formel 2.16 Produktionswahrscheinlichkeit	13
Formel 2.17 Forward-Wahrscheinlichkeit.....	13
Formel 2.18 Initial-Forward-Wahrscheinlichkeit	13
Formel 2.19 Forward-Algorithmus	14
Formel 2.20 Forward-Ergebnis.....	14
Formel 2.21 Wahrscheinlichster Pfad	14
Formel 2.22 Wahrscheinlichster Pfad pro Zustand	15
Formel 2.23 Viterbi Initialisierung.....	15
Formel 2.24 Viterbi-Algorithmus.....	15
Formel 2.25 Viterbi-Ergebnis.....	15
Formel 2.26 Neubelegung der Parameter der Übergangsmatrix	16
Formel 2.27 Backward-Wahrscheinlichkeit	17
Formel 2.28 Backward-Initialzustand	17
Formel 2.29 Backward-Algorithmus.....	17
Formel 2.30 A-posteriori-Wahrscheinlichkeit eines Zustands	17
Formel 2.31 A-posteriori-Wahrscheinlichkeit eines Zustandswechsel	17
Formel 2.32 Parameteranpassung mittels A-posteriori-Wahrscheinlichkeiten	18
Formel 2.33 Psojourn	21
Formel 2.34 Psojourn Berechnung	21
Formel 2.35 Pchange	21
Formel 2.36 Initialzustand der CHnMM	22
Formel 2.37 Folgezustände der CHnMM.....	22
Formel 2.38 CHnMM Evaluation Ergebnis	23

Formel 2.39 CHnMM Decodierung	23
Formel 2.40 CHnMM Decodierung Pfadauswahl	23
Formel 2.41 CHnMM Backward-Wahrscheinlichkeit	24
Formel 2.42 CHnMM pathProb	24
Formel 3.1 Übergangsmatrix der einer einzelnen Maschine	33
Formel 3.2 Modellüberführung	34
Formel 3.3 Gleichungssystem zur Bestimmung von Ausfall- und Reparaturrate	34
Formel 3.4 Dynamische Ausfallrate tbf_2	34
Formel 3.5 Psojourn der Maschinen stromabwärts	40
Formel 3.6 Pchange der Maschinen stromabwärts	41
Formel 3.7 Gleichungssystem zum Lösen der Outputmatrix-Übergänge	42
Formel 3.8 Faktor AAIA	43

1 EINLEITUNG

System Reliability Analysis ist eine eigenständige, interdisziplinäre Wissenschaft, die Methoden zur Einschätzung der Zuverlässigkeit von Systemen liefert und weit verbreitet Anwendung findet, da die Zuverlässigkeit insbesondere von technischen Systemen immer mehr an Bedeutung gewinnt. (vgl. Zio 2013, S. 7–8) Doch obwohl die Berechnung der Verfügbarkeit eines einfachen Systems auf ein Markovmodell beruht, werden diese besonders bei komplexen stochastischen Systemen kaum angewendet, da es in solchen Fällen schnell zu einem sehr großen Zustandsraum kommen kann, der die Analyse mittels Markovmodelle zu aufwendig werden lässt, weswegen dann computergestützte Simulation eingesetzt wird. (vgl. Krull 2008, S. 2)

Doch gerade bei stochastischen Systemen, die nur zum Teil beobachtbar sind, bieten sogenannte Hidden Markov Modelle interessante Möglichkeiten.

1.1 Ausgangslage

Hidden Markov Modelle sind bei der Spracherkennung und anderer Mustererkennung häufig angewandte Methoden, doch sie sind lediglich gedächtnislos und bieten daher wenig Realismus bei Anwendung auf technischen Systemen. (vgl. Fink 2008, S. 2–4)

Deswegen wurden die Hidden non-Markovian Models entwickelt, die auch andere Verteilungen als die Exponentialverteilung bei dem Modellieren des Systems zulassen. Eine Klasse dieser Modelle ist das Conversive Hidden non-Markovian Model. Dessen Veröffentlichung liegt jedoch noch nicht lange zurück. Somit ist es wenig erprobt.

1.2 Zielsetzung

Ziel dieser Arbeit ist es eine beispielhafte Anwendung der CHnMM bei der Zuverlässigkeitsanalyse eines technischen Systems durchzuführen und zu dokumentieren. Insbesondere liegt der Fokus dabei auf die Herausforderungen, die die Anwendung der CHnMM auf ein komplexes System mit sich bringt, und wie darauf reagiert wurde. Die Lösungswege werden aufgezeigt mitsamt ihrer Auswirkungen auf die Implementation des Models. Zum Abschluss folgte eine kurze Analyse der Analyseergebnisse der Anwendungsfälle Evaluation, Decodierung und Training.

1.3 Aufbau der Arbeit

Auf den Weg dahin wird zuerst im zweiten Kapitel eine kurze Einführung in relevante Informationen über die System Reliability Analysis gegeben. Im Anschluss wird das Konzept der Hidden Markovmodells vorgestellt und daraufhin das Konzept des Converseive Hidden non-Markovian Model.

Im dritten Kapiteln folgt die Fallstudie selbst. Zuerst wird die Pizza-Produktionslinie vorgestellt, die in dieser Fallstudie analysiert wird. Dann folgt die Beschreibung der Entwicklung des Hidden Markov Model der Produktionslinie, die als Benchmark dient. Schließlich folgt der Kernpunkt der Arbeit, die Entwicklung des Converseive Hidden non-Markovian Model und im Anschluss daran die Diskussion der Ergebnisse der Analyse der Produktionslinie durch die Markovmodelle.

2 GRUNDLAGEN

Ein System ist, allgemein betrachtet, eine „abgegrenzte Anordnung einer Menge von Elementen und einer Menge von Relationen zwischen diesen Elementen“ (vgl. Gottschalk 1971, S. 258) Zudem werden in dieser Arbeit ausschließlich dynamisch technische Systeme betrachtet, die aus Ausschnitten der materiellen Welt bestehen, die alle für die Funktionalität des Systems notwendigen Elemente enthalten. (vgl. Trömel 1995, S 161) Dabei ist die Funktion eines technischen Systems dessen gezielter, materieller Einfluss auf die Umwelt. (vgl. Gottschalk 1971, S. 260)

2.1 System Reliability Analysis

System Reliability Analysis dient zur Bewertung der Ausfallwahrscheinlichkeit eines Systems mit Hilfe einer Sammlung formaler Methoden zur Untersuchung der verschwommenen Grenzen zwischen einem korrekt operierenden System und einem Ausfall. (vgl. Zio 2013, S. 59) Sie entwickelte sich seit dem Ende des 2. Weltkrieges zu einer eigenständigen Wissenschaft, um die häufigen Ausfälle technischer militärischer Ausrüstung zu untersuchen, wobei die Forschung zunächst auf einzelne Komponenten von Systemen beschränkt wurde. In den 1960er richtete sich der Fokus der System Reliability Analysis dann stärker auf ganze Systeme, da diese zunehmend komplexer wurden und sich aus einer größeren Anzahl an Komponenten zusammensetzten. Zudem kristallisieren sich spezielle Methoden und Anwendungsfelder heraus. (vgl. Zio 2013, S.7-9)

Aufbauend auf den Methoden der Wahrscheinlichkeitstheorie und Statistik wird die System Reliability Analysis technischer Systeme durch Verfahren der diskreten Mathematik und mathematischen Optimierung erweitert, um die Zuverlässigkeit (engl. reliability) technischer Systeme zu bewerten, zu prognostizieren, zu erhalten und zu verbessern. Die zahlreichen Ursachen für die Ausfälle technischer Systeme werden beschrieben und erfasst um Maßnahmen zu treffen, mit dem Ziel die Auswirkungen der Ausfälle zu begrenzen. (vgl. Ryll 2011, S.43-44)

2.1.1 Zuverlässigkeit technischer Systemen

Bei der *Zuverlässigkeit* handelt es sich um die Wahrscheinlichkeit, dass das betrachtete System für die Dauer eines gegebenen Zeitintervalls durchgängig einsatzbereit ist. (vgl. Koether 2007, S. 198)

Weichen nominale Eigenschaften eines Systems aufgrund einer Wertminderung auf nicht akzeptable Art und Weise von den geforderten Eigenschaften ab, so dass die Komponente nicht mehr die ihr angedachte Funktion erfüllen kann, kommt es zu einem Ausfall. Demnach ist ein Ausfall der Übergang des System oder einer Komponente von einem Zustand sachgemäßer Funktionalität zu einer Störung. Schränkt die Störung die sachgemäße Funktionalität des Systems komplett ein, handelt es sich um einen kritischen Ausfall. Ist die korrekte Ausübung der Funktion nur teilweise nicht gegeben, liegt ein Ausfall mit primärer Bedeutung vor. Ist die sachgemäße Funktionalität vom Werteverlust der Systemkomponente nicht betroffen, kommt dem Ausfall nur eine sekundäre Bedeutung zu. (vgl. Lazzaroni 2007, S.4-5)

Dabei steht die Zuverlässigkeit eines Systems in Relation zu anderen quantifizierbaren Kenngrößen, durch die das dynamische Verhalten der Systemfunktionalität erfasst werden können. Diejenigen, auf die sich diese Arbeit bezieht, sind wie folgt:

Die *Ausfallwahrscheinlichkeit* bezeichnet die Wahrscheinlichkeit, ein System oder dessen Komponente in einem vorgegebenen Intervall ausfällt. Die *Ausfallrate* steht für die Rate, mit der ein System oder deren Komponente zu einem bestimmten Zeitpunkt ausfällt, wenn es bis dahin korrekt arbeitete. Die *mittlere Lebensdauer* besteht aus dem Erwartungswert der Zeitdauer zwischen der Aufnahme der Funktionalität bis zur Einstellung dieser durch eine Störung. Bei der Verwendung von einfachen Systemen bei der System Reliability Analysis steht somit die *Betriebszeit* des technischen Systems, die der Differenz des *Zeitverlustes*, der durch kritische Ausfälle verursacht wird, von der geplanten Produktionszeit entspricht, im Fokus. (vgl. Ryll 2011, S. 81). Bei der Betrachtung der Zuverlässigkeit technischer Systeme wird zwischen einfachen und strukturierten Vertretern unterschieden. (vgl. Ryll 2011, S. 44)

2.1.2 Zuverlässigkeit einfacher Systeme

„Ein einfaches System ist dadurch gekennzeichnet, dass es hinsichtlich des Ausfallverhaltens und der Durchführung von Instandhaltungsmaßnahmen nicht weiter differenziert werden kann und selbst nicht instandsetzbar ist, wie z.B. Bauteile.“ (Ryll 2011, S. 45)

Unter der Voraussetzung, dass solche Systeme lediglich zwei diskrete Zustände annehmen können, kann der Systemzustand durch die Boolesche Variable $z(t)$ erfasst werden. Deren mögliche Belegung lautet wie folgt: *funktionstüchtig* - dann ist $z(t) = 1$, d.h. das System

nimmt die geforderte Funktionalität wahr -, und *funktionsuntüchtig*. - dann ist $z(t) = 0$, d.h. das System erfüllt die angedachte Funktionalität nicht- Zudem sind sie zu Anfang des betrachteten Zeitintervalls funktionstüchtig und können, nachdem sie einmal durch einen kritischen Ausfall funktionsuntüchtig wurden, nicht wieder in den vorherigen Zustand wechseln. In Abhängigkeit einer bestimmten Lebensdauer L und der Zeitdauer t lässt sich der Systemzustand wie folgt beschreiben (vgl. Ryll 2011, S. 45):

$$z(t) = \begin{cases} 1, & t < L \\ 0, & t \geq L \end{cases}$$

Formel 2.1 Zustand eines Systems



Abbildung 2.1 Zustandsänderung von einfachen Systemen ohne Instandsetzung (vgl. Ryll 2011, S. 45)

Hierbei handelt es sich um eine stochastischen Prozess, da ein Wahrscheinlichkeitsmodell vorliegt, das die zufällige Entwicklung des Systems über die Zeit beschreibt (vgl. Kulkarni, 2011, S. 1). In dem Modell entspricht die Ausfallwahrscheinlichkeit $F(t)$ dem Wert der Verteilungsfunktion der stochastischen Lebensdauer des Systems für den Zeitpunkt t (vgl. Ryll 2011, S. 45). Der Erwartungswert dieser Verteilungsfunktion stellt die mittlere Lebensdauer dar (vgl. Ryll 2011, S. 49).

$$F(t) = P(L < t)$$

Formel 2.2 Verteilungsfunktion

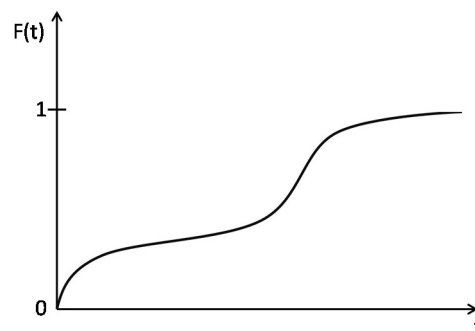


Abbildung 2.2 Beispielhafter Verlauf der Ausfallwahrscheinlichkeit $F(t)$ (vgl. Ryll 2011, S. 46)

Diese Funktion, deren Wertebereich zwischen 0 und 1 liegt und die Wahrscheinlichkeit P berechnet, mit der im Zeitintervall $[0, t)$ ein kritischer Ausfall stattfindet, ist monoton steigend, wobei der Wert für den Zeitpunkt $t=0$ zum Start des betrachteten Intervalls - Null entspricht, da zu Beginn mit keinem Ausfall zu rechnen ist, und für t gegen Unendlich nach Eins strebt, da ein Ausfall irgendwann auf jeden Fall auftritt. (vgl. Ryll 2011, S. 46) Somit gilt:

$$F(0) = 0$$

$$\lim_{t \rightarrow \infty} F(t) = 1$$

$$0 \leq F(t) \leq 1$$

Formel 2.3 Eigenschaften der Verteilungsfunktion

Die Zuverlässigkeit $R(t)$ ist die Gegenwahrscheinlichkeit zur Ausfallwahrscheinlichkeit. Zudem entspricht sie dem Erwartungswert des Systemzustandes $z(t)$. (vgl. Ryll 2011, S. 46-49)

$$R(t) = 1 - F(t)$$

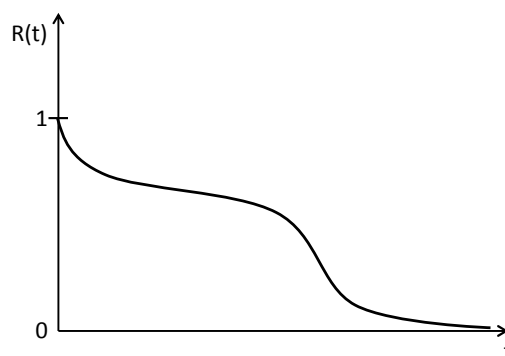


Abbildung 2.3 Beispielhafter Verlauf der Zuverlässigkeit $R(t)$ (vgl. Ryll 2011, S. 47)

Abgeleitet von der Ausfallwahrscheinlichkeit, quantifiziert die *Ausfalldichte* $f(t)$ die Wahrscheinlichkeit für einen Systemausfall innerhalb eines kleinen Zeitraums $(t, t + \Delta t)$, wobei Δt gegen 0 strebt. (vgl. Ryll 2011, S. 47)

$$f(t) = \frac{dF(t)}{dt}$$

Formel 2.4 Ausfalldichte

Der Wert der Ausfalldichte berücksichtigt allerdings nicht den Verlauf der Ausfallwahrscheinlichkeit vor dem Zeitpunkt t . Um die Ausfallrate $\lambda(t)$ zu ermitteln, wird daher die Ausfalldichte durch die Zuverlässigkeit geteilt. (vgl. Ryll 2011, S. 48)

$$\lambda(t) = \frac{f(t)}{R(t)} = \frac{f(t)}{1 - F(t)}$$

Formel 2.5 Ausfallrate

Bei technischen Systemen ist die Ausfallrate in der Regel nicht konstant. Vielmehr nimmt sie von der erstmaligen Inbetriebnahme des Systems bis zum Ende der Nutzungsdauer idealisiert die Form einer Badewanne an. (vgl. Ryll 2011, S. 48)

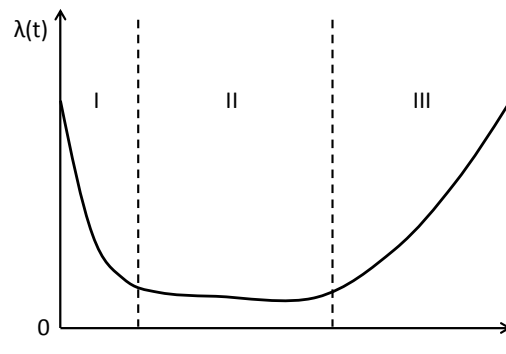


Abbildung 2.4 Beispielhafter Verlauf der Ausfallrate $\lambda(t)$ (vgl. Ryll 2011, S. 48)

Aus dieser Darstellung lassen sich drei Phasen mit unterschiedlichen dynamischen Verhalten der Ausfallrate identifizieren. Ausfälle können daher auch durch die Phase im Lebenszyklus der Systemkomponente charakterisiert werden, in der sie auftreten, was Spekulationen auf die Ursache eines Ausfalls zulässt. In der Phase I, der Phase der Frühausfälle während oder kurz nach Aufnahme der Funktionalität, sind vor allem Fertigungs-, Material- und Montagefehler der Systemkomponenten für Störungen verantwortlich. Zudem kommt es aufgrund der Einarbeitung des Personals an der neuen Komponente vermehrt zu Bedienungsfehlern. In der Phase II, der Phase des konstanten Betriebs, hingegen kommt es meist allenfalls zu zufälligen Störungen, da die Frühausfälle nicht mehr vorkommen und Verschleißteile im Zuge regelmäßige Wartungen ersetzt werden. Erst wenn diese die Ermüdung und die Alterung des Systems oder dessen Komponenten nicht mehr kompensieren können, kommt es vermehrt zu Spätausfällen, womit die letzte Phase, die Phase III erreicht wäre. (vgl. Arnold 2008, S 865-866)

2.1.3 Zuverlässigkeit strukturierter Systeme

Strukturierte Systeme bestehen aus einer bestimmten Anzahl N an einfachen Systemen mit voneinander unabhängigen stochastischen Lebensdauern L_n und eigenen Ausfallwahrscheinlichkeiten $F_n(t)$. Zudem sind die Zustände der einzelnen Komponenten der Struktur untereinander nicht verknüpft, sodass der Zustand eines einfachen Systems

keinen Einfluss auf den Ausfall oder das Überleben eines anderen Systems hat, nur auf das strukturierte System selbst. Bei einer System Reliability Analysis lassen sich die Grundsysteme in zwei typische Arten unterteilen. Parallelsysteme, die erst ihre sachgemäße Funktionalität nicht erfüllen können, wenn alle Systemkomponenten ausfallen, werden in dieser Arbeit jedoch nicht näher betrachtet. Stattdessen wird im Folgenden näher auf die *Seriensysteme* eingegangen. Diese fallen aus, wenn auch nur eine Komponente ihre Funktion nicht mehr ausüben kann, da sie in Reihe geschaltet sind. (vgl. Ryll 2011, S. 59-60) E₁

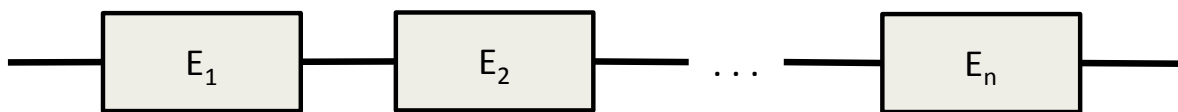


Abbildung 2.5 Schaltbild eines Seriensystems (Ryll 2011, S. 60)

Die Zuverlässigkeit R_S des Seriensystems, die auch in diesem Fall der Verfügbarkeit gleichzusetzen ist, berechnet sich aus dem Produkt der Zuverlässigkeit aller Komponenten. (vgl. Ryll 2011, S. 60)

$$R_S(t) = \prod_{n=1}^N R_n(t)$$

Formel 2.6 Zuverlässigkeit eines Seriensystems (vgl. Ryll 2011, S. 60)

Die Ausfallwahrscheinlichkeit des Seriensystems ergibt aus der Relation zu dessen Zuverlässigkeit. (vgl. Ryll 2011, S. 60)

$$F_S(t) = 1 - R_S(t)$$

Die Ausfallrate λ_S des Seriensystems entspricht der Summe der Ausfallraten der einzelnen Komponenten. (vgl. Ryll 2011, S. 60)

$$\lambda_S(t) = \sum_{n=1}^N \lambda_n(t)$$

Formel 2.7 Ausfallrate eines Seriensystems

2.1.4 Die Verfügbarkeit instandsetzbarer Systeme

Bei Systemen, die sich nach einer Störung instand setzen lassen und wieder ihre Funktion erfüllen können, verfügt die Zuverlässigkeit nur über begrenzte Aussagekraft, da sie als Gegenwahrscheinlichkeit zur Ausfallwahrscheinlichkeit weder die Dauer einer Störung noch die Dauer darauffolgender Reparatur berücksichtigt. An dieser Stelle setzt die *Verfügbarkeit* $V(t)$ an, als Bezeichnung für den Anteil der Zeit, in der das System

einsetzbar ist, gegenüber der Zeit, in der es einsetzbar sein soll. (vgl. Lazzaroni 2011, S. 85)

Zu der Zeit bis zum Ausfall kommen bei instandsetzbaren Systemen die Zeitintervalle zwischen Ausfällen an Betriebszeit hinzu. Nach einem Ausfall steht das System für eine Zeitspanne nicht zur Verfügung, in der die Funktionalität des Systems wiederhergestellt wird. Demnach bezeichnet die *Wartbarkeit* die Wahrscheinlichkeit, dass das System innerhalb eines vorgegebenen Intervalls repariert wird, und die Reparaturrate steht für die Wahrscheinlichkeit, dass ein System in einem bestimmten Augenblick repariert wird, sofern es denn seit dem vorherigen Ausfall nicht funktionstüchtig geworden ist. (vgl. Lazzaroni 2011, S. 86)

Vereinfacht können selbst komplexe technische Systeme über ein binäres Markovmodell abgebildet werden, dessen Zustandsgraph in folgender Abbildung dargestellt wird. (vgl. Arnold 2007, S. 303)

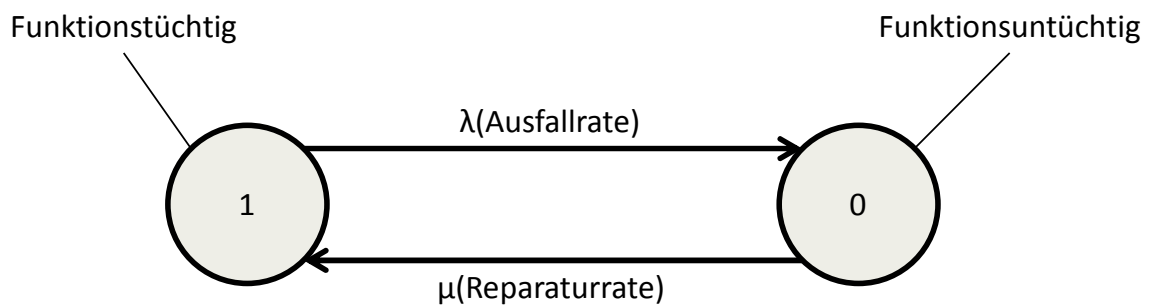


Abbildung 2.6 Zustandsgraph eines binären technischen Systems (vgl. Arnold 2007, S. 303)

Das Markov Modell des technischen Systems setzt sich dabei aus den zwei bereits bekannten Zuständen funktionstüchtig = 1 und funktionsuntüchtig = 0 zusammen. Der Austausch an Wahrscheinlichkeit, dass das System sich in jenen Zustand im betrachteten Augenblick befindet, wird durch die Ausfallrate λ und die Reparaturrate μ bestimmt. Unter den Voraussetzungen, dass die Übergangsraten zeithomogen sind, das System sich jederzeit in einen der beiden Zustände befindet und lediglich der gegenwärtige Zustand des Markovmodells einen Einfluss auf den zukünftigen hat, kann auf folgende Gleichung geschlossen werden, die die Verfügbarkeit v des Systems für t gegen Unendlich berechnet: (vgl. Arnold 2007, S. 304)

$$v = \frac{\mu}{\mu + \lambda}$$

Formel 2.8 Verfügbarkeit eines Systems

Die Anwendbarkeit des binären Markovmodells ist jedoch begrenzt. Über dieses Modell lässt sich die beobachtete oder erwartete Verfügbarkeit eines technischen Systems nur quantifizieren, die Ursachen für das Ergebnis werden jedoch nicht untersucht. Im Falle eines komplexen Systems wirken viele Elemente, die wiederum Subsysteme sein können, aufeinander (vgl. Gottschalk 1971, S. 260). Um Handlungsempfehlungen aus der System Reliability Analysis ziehen zu können, müssen insbesondere komplexe Systeme auf einen niedrigeren Abstraktionsgrad betrachtet werden. Dies schließt Markovmodelle als Werkzeug jedoch nicht aus, da diese nicht binär sein müssen.

Besteht das System aus zwei verschiedenen Subsystemen, von denen jedes wiederum nur die Zustände funktionstüchtig und funktionsuntüchtig annehmen kann, so lässt sich dieses mit Hilfe eines Markovmodells mit vier Zuständen darstellen. Steht X_1 und X_2 jeweils für die funktionstüchtigen Zustände von System Eins und Zwei, sowie Y_1 und Y_2 für deren funktionsuntüchtiges Verhalten, so folgt daraus folgender Zustandsraum S_i (vgl. Lazzaroni 2011, S. 101)

$$S_i == \begin{cases} S_0 = X_1 X_2 \\ S_1 = X_1 Y_2 \\ S_2 = Y_1 Y_2 \\ S_3 = Y_1 X_2 \end{cases}$$

Formel 2.9 Zustandsraum eines Zweimaschinenmodells

Wird auf diese Art und Weise ein Markov Modell über mehrere Systeme zusammengestellt, kommt es jedoch zu einem exponentiellen Wachstum des Zustandsraums, da das Modell bei N Elementen des betrachteten Systems bis zu 2^N Zustände haben kann (vgl. Lazzaroni 2011, S. 104). Da technische Systeme jedoch eher an Komplexität zunehmen, stellt der Zustandsraum der Modelle eine Herausforderung dar. Einen Weg ihr zu begegnen, wird später im Rahmen dieser Arbeit vorgestellt.

2.2 Hidden Markov Models

Nicht immer lässt sich das dynamische Verhalten eines Systems direkt beobachten, sondern nur teilweise oder indirekt. Sendet das System jedoch Symbole aus, die Rückschlüsse auf die internen Vorgänge zulassen, gibt es Methoden diese systematisch auszuwerten. (vgl. Buchholz 2012, S. 8)

Eine solche Methode der Markov Prozesse, die es nicht nur ermöglicht, das Verhalten eines Systems möglichst genau zu erfassen, sondern auch das beobachtete Verhalten eines Systems zu segmentieren, ist das Hidden Markov Model (HMM). Ein solches besteht aus zwei Pfeilern. Der eine Pfeiler ist im Zuge dieser Arbeit eine Discrete Time Markov Chain, der andere besteht aus einer Sequenz an Outputsignalen, eines für jeden betrachteten Zeitpunkt. (vgl. Fink 2008, S. 62-63)

2.2.1 Discrete Time Markov Models

Eine Discrete Time Markov Chain (DTMC) bilden das dynamische Verhalten einer HMM ab. Dabei handelt es sich um einen stochastischen Prozess $\{X_t, t \geq 0\}$ auf dem endlichen Zustandsraum $S = \{S_1, S_2, \dots, S_N\}$, bei dem der Zustand eines Systems X_{t+1} zum Zeitpunkt $t+1$ auf Grundlage der Gegenwart in Form von X_t abgeleitet werden kann. Dass die Vergangenheit des Systems keinen Einfluss auf die Zukunft hat, sondern nur die Gegenwart, womit das System ohne Gedächtnis ist, wird als Markoveigenschaft bezeichnet. (vgl. Kulkarni 2011, S. 5)

Daher gilt:

$$P(X_{t+1} = S_j | X_t = S_i, X_{t-1}, \dots, X_0) = P(X_{t+1} = S_j | X_t = S_i) \forall S_i, S_j \in S$$

Formel 2.10 Markoveigenschaft eines Modells

Zudem wird der Prozess als Zeit homogen angesehen.

$$P(X_{t+1} = S_j | X_t = S_i) = P(X_1 = S_j | X_0 = S_i) \forall S_i, S_j \in S$$

Formel 2.11 Zeithomogenität

Somit ist die bedingte Wahrscheinlichkeit für den Übergang von einem Zustand i in den Zustand j während einer Zeiteinheit unabhängig vom betrachteten Zeitpunkt. Diese bedingte Wahrscheinlichkeit wird als *schrittweise Übergangswahrscheinlichkeit* bezeichnet und mit $a_{i,j}$ abgekürzt. (vgl. Kulkarni 2011, S. 6)

$$A = \{a_{i,j} | a_{i,j} = P(X_{t+1} = S_j | X_t = S_i)\} \forall S_i, S_j \in S$$

Diese Werte lassen sich in einer $N \times N$ Matrix, der Übergangsmatrix A , zusammenfassen. Jeder Eintrag dieser Matrix steht für eine schrittweise Übergangswahrscheinlichkeit, wobei die Zeile den Sender und die Spalte den Empfänger der bedingten Wahrscheinlichkeit identifiziert, und dessen Wert ist größer gleich Null. Zudem entspricht die Summe aller Einträge einer Zeile Eins. (vgl. Kulkarni 2011, S. 6-7)

Der Zustand der DTMC im Zeitpunkt t wird über den Zeilenvektor $\boldsymbol{\pi}_t = [\pi_t(S_1), \pi_t(S_2), \dots, \pi_t(S_N)]$ festgehalten, wobei gilt:

$$\pi_t(S_i) = P(X_t = S_i) \forall S_i \in S$$

Formel 2.12 Systemzustand

Berechnet wird $\boldsymbol{\pi}_{t+1}$ ausgehend von dem Zustand der DTMC zum Zeitpunkt t , der folgenden Gleichung entsprechend (vgl. Kulkarni 2011, S. 15-16):

$$\boldsymbol{\pi}_{t+1} = \boldsymbol{\pi}_t * A$$

Formel 2.13 DTMC Schritt von t nach $t+1$

Ausgehend von dem Initialzustand in Form des Vektors $\boldsymbol{\pi}_0$, dessen Einträge in der Summe Eins ergeben, und der Annahme der Zeithomogenität folgt (vgl. Kulkarni 2011, S. 16):

$$\boldsymbol{\pi}_t = \boldsymbol{\pi}_0 * A^t$$

Formel 2.14 DTMC Schritt von 0 nach t

2.2.2 Spezifikation eines Hidden Markov Model

Im Rahmen dieser Arbeit wird für die Sequenz an Outputsignalen, der Beobachtungssequenz, ein diskretes, finite Menge an Symbolen $V = \{V_1, V_2, \dots, V_L\}$ an möglichen Signalen vorausgesetzt, dessen Elemente sich zu einer Reihe an Sequenz an Symbolen $O = O_1, O_2, \dots, O_T \in V$ zusammensetzen, die über einen begrenzten Zeitraum, unterteilt in T Intervalle, erfasst wurden. Diese gehen einher mit Zustands spezifischen Wahrscheinlichkeiten $b_{i,k}$, die zu einer Outputmatrix zusammengefasst werden, welche wie folgt definiert ist (vgl. Fink 2008, S. 63):

$$B = \{b_{i,k} | b_{i,k} = P(O_t = V_k | X_t = S_i)\} \forall S_i \in S, V_k \in V$$

Formel 2.15 Outputmatrix

Demzufolge ist der Eintrag $b_{i,k}$ die Wahrscheinlichkeit, dass der Zustand S_i zu einem Zeitpunkt t das Symbol $V_k \in V$ sendet. Fügt man diese Outputmatrix mit der DTMC zusammen, ergibt sich ein Fünftupel aus der Menge an Zuständen S , der Menge an möglichen Outputsignalen V , der Übergangsmatrix A , Outputmatrix B und dem Initialzustand $\boldsymbol{\pi}_0$, welches das HMM eindeutig beschreibt. Dabei spezifiziert das Parameterset $M = (A, B, \boldsymbol{\pi}_0)$ das eigentliche Modell des betrachteten Systems, mitsamt seines dynamischen Verhaltens. (vgl. Rabiner 1989, S. 260–261)

In Kombination von M mit den Parametern S und V zu einem HMM können jedoch weit reichendere Fragestellungen gelöst werden, als nur das dynamische Verhalten des Systems. Diese Problemstellungen lassen sich in drei grundlegende Anwendungsfälle einordnen:

Der Evaluation, der Decodierung und dem Training. Diese werden in den folgenden drei Abschnitten mitsamt den etablierten Lösungsverfahren näher beleuchtet. (vgl. Fink 2008, S. 66)

2.2.3 Anwendungsfall Evaluation

Dem Anwendungsfall der Evaluation liegt die Frage zu Grunde, wie wahrscheinlich es ist, dass eine beobachtete Outputsequenz O von dem Model M erzeugt werden konnte. Gesucht ist demnach die *Produktionswahrscheinlichkeit* $P(O|M)$. Ein nicht praktikabler Lösungsansatz wäre es, die Wahrscheinlichkeit aller möglichen Zustandssequenzen $Q = Q_1, Q_2, \dots, Q_T \in S$, die das Model durchlaufen kann, zu berechnen und zu addieren, wie die folgende Gleichung beschreibt. (vgl. Fink 2008, S.68–69)

$$\sum_Q P(O|Q, M) = \sum_{sQ} \prod_{t=1}^T a_{Q_{t-1}, Q_t} b_{Q_t}(O_t)$$

Formel 2.16 Produktionswahrscheinlichkeit

Der Aufwand von $O(TN^T)$ ist jedoch zu hoch. Verwendet wird stattdessen der *Forward-Algorithmus*. Dieser nutzt die Eigenschaft der Gedächtnislosigkeit von Markovketten aus. Da aus ihr folgt, dass für das zukünftige Verhalten des Models nur der gegenwärtige Zustand t von Bedeutung ist, kann der Modellzustand zum Zeitpunkt t ohne ermittelt werden, ohne dass der bisherige Pfad zur Verfügung stehen muss. Demzufolge wird die *Forward-Wahrscheinlichkeit* $\alpha_t(S_i)$ eingeführt, deren Wert für die Wahrscheinlichkeit steht, dass durch das Modell M die bisherige bis O_t generiert wurde und es sich zum Zeitpunkt t im Zustand S_i befindet. Somit gilt: (vgl. Fink 2008, S. 69–70)

$$\alpha_t(S_i) = P(O_1, O_2, \dots, O_t, Q_t = S_i | M)$$

Formel 2.17 Forward-Wahrscheinlichkeit

Initiiert wird der Forward-Algorithmus im Zeitpunkt 1, wobei der Forward-Wert für jedes i von 1 bis N über folgende Gleichung bestimmt wird. (vgl. Fink 2008, S. 71)

$$\alpha_1(S_i) = \pi_0(S_i) b_i(O_1)$$

Formel 2.18 Initial-Forward-Wahrscheinlichkeit

Der Initialisierung des Verfahrens folgt die Berechnung der Forward-Werte entlang der Zeitstrecke, wobei für einen Zeitschritt nur die Werte aus dem vorherigen abrufbar sein müssen. Folgende Gleichung zeigt dabei die Vorgehensweise. (vgl. Fink 2008, S. 71)

$$\alpha_{t+1}(S_j) = \left(\sum_{i=1}^N \alpha_t(S_i) a_{i,j} \right) b_j(O_{t+1})$$

Formel 2.19 Forward-Algorithmus

Zum Ende des Forward-Algorithmus hin stehen N Forward-Werte des Zeitpunkts T zur Verfügung, deren Summe der Produktionswahrscheinlichkeit $P(O|M)$ entspricht. (vgl. Fink 2008, S. 71)

$$P(O|M) = \sum_{i=1}^N \alpha_T(S_i)$$

Formel 2.20 Forward Ergebnis

Anhand derer kann im direkten Vergleich aus einer Menge von Modellen jenes identifiziert werden, das die beobachtete Sequenz O am ehesten erzeugt. (vgl. Fink 2008. S. 4)

2.2.4 Anwendungsfall Decodierung

Die zweite Problemstellung beschäftigt sich mit der Frage, welche optimale Sequenz an Zuständen Q des Modells M die Sequenz an beobachteten Symbolen O am besten erklärt. Es wird dementsprechend versucht, von Q auf O zu schließen. (vgl. Rabiner 1989, S. 261)

Gesucht wird demnach die Zustandssequenz, die in Kombination mit einem spezifischen Modell und einer beobachteten Sequenz an Symbolen die höchste Wahrscheinlichkeit ergibt. Auch hier wäre die naive Variante, siehe Formel 2.21, die Wahrscheinlichkeit aller möglichen Sequenzen zu berechnen und die Sequenz mit dem höchsten Wert zu identifizieren, kaum praktikabel, weil auch für diesen Anwendungsfall der einfache Lösungsweg eine exponentielle Komplexität von $O(TN^T)$ aufweist. (vgl. Fink 2008, S. 74–75)

$$Q^* = \underset{Q}{\operatorname{argmax}} P(Q|O, M) = \underset{Q}{\operatorname{argmax}} P(O, Q|M)$$

Formel 2.21 Wahrscheinlichster Pfad

Wie bei der Evaluation kann auch in diesem Fall die Gedächtnislosigkeit der HMM ausgenutzt werden. Da der vergangene Pfad an Zuständen für den zukünftigen keine Rolle spielt, lassen sich alle Pfade, die in der Gegenwart t zum Zustand $S_i = Q_t$ führen, als Ergebnis ausschließen, deren momentane Wahrscheinlichkeit nicht der maximalen des betrachteten Zeitpunktes wenigstens entspricht. Dieser Ansatz führt zum *Viterbi-Algorithmus* Pro Zustand $S_i \in S$ wird nur ein Pfad mit der maximalen Wahrscheinlichkeit

$\delta_t(S_i)$ für den Zeitpunkt t ausgewählt, wie folgende Gleichung erläutert. (vgl. Fink 2008, S. 73)

$$\delta_t(S_i) = \max_{Q_1, Q_2, \dots, Q_t} P(O_1, O_2, \dots, O_t, Q_1, Q_2, \dots, Q_{t-1}, Q_t = S_i | M)$$

Formel 2.22 Wahrscheinlichster Pfad pro Zustand

Analog zum Forward-Algorithmus in Abschnitt 2.2.3 wird auch dieses Verfahren zum Zeitpunkt 1 über alle $S_i \in S$ initialisiert, getreu der folgenden Gleichung. (vgl. Fink 2008, S. 73)

$$\delta_1(S_i) = \pi_0(S_i) b_i(O_1)$$

Formel 2.23 Viterbi Initialisierung

Eine weitere Parallele zum Forward-Verfahren ist die Berechnung der Wahrscheinlichkeiten der Teilpfade aller Zustände entlang der Zeitstrecke. Der einzige Unterschied besteht in der Suche nach dem Pfad mit der höchsten Wahrscheinlichkeit, statt der Aufsummierung der Wahrscheinlichkeiten aller möglichen Pfade für den betrachteten Zustand, wie es die folgende Gleichung demonstriert. (vgl. Fink 2008, S. 73)

$$\delta_{t+1} = \max_i \{ \delta_t(S_i) a_{i,j} \} b_j(O_{t+1})$$

Formel 2.24 Viterbi-Algorithmus

Dieser Unterschied bleibt zum Zeitpunkt T bestehen, so dass auch in diesem nach dem Pfad mit der größten Wahrscheinlichkeit über alle Zustände identifiziert wird. Siehe folgende Gleichung. (vgl. Fink 2008, S. 76)

$$Q_T^* = \operatorname{argmax}_i \delta_T(S_i)$$

Formel 2.25 Viterbi Ergebnis

Im Anschluss an den Viterbi-Algorithmus bedarf es jedoch einer Möglichkeit, den Verlauf des identifizierten Pfades zurückzuverfolgen, um die Decodierung abschließen zu können. (vgl. Fink 2008, S. 76)

2.2.5 Anwendungsfall Training

Die Anwendung von HMM erfordert Modelle, die die statistischen Eigenschaften von betrachteten Daten so nah wie möglich wiedergeben. Jedoch ist keine Methode zu bekannt, mit der aus einem Datensatz auf ein Modell geschlossen werden kann. Daher bedarf es Experten, die mit ihrem Vorwissen wenigstens die Struktur des gesuchten Modells erstellen, mitsamt der Outputmatrix und einer sinnvoller Schätzungen aller Parameter. Ist

dieser Schritt getan, kann jedoch das Modell Schritt für Schritt optimiert werden, was den Anwendungsfall des Trainings darstellt. (vgl. Fink 2008, S. 76)

Dabei gibt es verschiedene Varianten des Trainings, die sich vor allem durch die Art der Bewertung des betrachteten Modells ergeben. Während dafür beim Viterbi-Training die höchste Pfadwahrscheinlichkeit verwendet wird, wie sie durch den Viterbi-Algorithmus berechnet wird, bezieht sich der Baum-Welch-Algorithmus auf die Evaluation durch den Forward-Algorithmus. Unabhängig vom Gütekriterium wird durch die in diesem Abschnitt vorgestellte Trainingsmethode mit jedem Optimierungsdurchlauf eine Verbesserung des Qualitätswertes des Modells erzielt, wobei die Verbesserungsrate irgendwann jedoch so klein werden kann, dass sie nicht mehr messbar ist und daher die Modellqualität auf einem Level verbleibt. Somit verhält sich die Modellqualität über ein Optimierungsschritt hinweg monoton steigend. (vgl. Fink 2008, S. 76–77)

Grundlage für das Training ist dabei die schrittweise Anpassung der Modellparameter auf Grundlage der Beobachtung des Verhaltens des Modells M angesichts einer Sequenz an ausgegebenen Symbolen O , wobei jedoch nur auf die zu erwartenden Ereignissen in Form von Zustandsübergängen und dem Senden von Symbolen zurückgegriffen werden kann. Im Rahmen dieser Arbeit wird jedoch nur die Optimierung der Einträge der Übergangsmatrix A betrachtet. Deren optimierte Werte $\hat{a}_{i,j}$ werden nach Observierung des Verhaltens des alten Modells in den Intervallen 1 bis T wie folgt berechnet: (vgl. Fink 2008, S. 77)

$$\hat{a}_{i,j} = \frac{\text{zu erwartende Anzahl an Übergängen von } S_i \text{ nach } S_j}{\text{zu erwartende Anzahl an Übergängen aus } S_i \text{ hinaus}}$$

Formel 2.26 Neubelegung der Parameter der Übergangsmatrix

Um die zu erwartende Anzahl an Übergängen des zu optimierenden Modells zu berechnen, bedarf es der Möglichkeit zur Auswertung der *A-posteriori-Wahrscheinlichkeit* eines Zustand oder eines Zustandsübergangs. Diese Wahrscheinlichkeit sagt aus, wie wahrscheinlich es ist, dass ein bestimmtes Modell zu einem bestimmten Zeitpunkt sich betrachteten Zustand befindet oder den betrachteten Zustandswechsel vollzieht. Im Falle eines Viterbi- Trainings kann diese direkt über den Pfad, der als Qualitätsmesser verwendet wird, abgelesen werden. Wird das Baum-Welch-Verfahren verwendet, wird zusätzlich zu der Forward-Wahrscheinlichkeit $\alpha_t(S_i)$ die *Backward-Wahrscheinlichkeit* $\beta_t(S_i)$ benötigt, die für die Wahrscheinlichkeit steht, dass vom Zustand S_i aus die

Teilsequenz $O_{t+1}, O_{t+2}, \dots, O_T$ durch das Modell M erzeugt wird. Somit gilt für $\beta_t(S_i)$: (vgl. Fink 2008, S. 77–78)

$$\beta_t(S_i) = P(O_{t+1}, O_{t+2}, \dots, O_T | Q_t = S_i, M)$$

Formel 2.27 Backward-Wahrscheinlichkeit

Berechnet werden die Backward-Wahrscheinlichkeit im Anschluss des Forward-Algorithmus, der einen Trainingslauf initiiert und die Produktionswahrscheinlichkeit, wie auch die Forward-Wahrscheinlichkeiten liefert, weswegen dieser Abschnitt auch *Forward-Backward-Algorithmus* heißt. Dabei werden die Backward-Wahrscheinlichkeiten im Fall des Endzeitpunkts T auf 1 festgelegt. Somit gilt: (vgl. Fink 2008, S. 79)

$$\beta_T(S_i) = 1$$

Formel 2.28 Backward-Initialzustand

Wie beim Forward- Algorithmus, Abschnitt 2.2.3, werden auch die Backward-Wahrscheinlichkeiten nach der Reihenfolge bestimmt, die die betrachtete Zeitstrecke vorgibt, mit Besonderheit, dass diese rückwärts abgearbeitet wird. Die Backward-Wahrscheinlichkeiten ergeben sich dann aus folgender Gleichung: (vgl. Fink 2008, S. 79)

$$\beta_t(S_i) = \sum_{j=1}^N a_{i,j} b_j(O_{t+1}) \beta_{t+1}(S_j)$$

Formel 2.29 Backward-Algorithmus

Das Produkt der Forward- und Backward-Wahrscheinlichkeit des Zustands S_i zum Zeitpunkt t , genormt durch die Produktionswahrscheinlichkeit $P(O|M)$ aus Abschnitt 2.2.3 ergibt die A-posteriori-Wahrscheinlichkeit γ_t des Zustands S_i , siehe folgende Gleichung. (vgl. Fink 2008, S. 78)

$$\gamma_t(S_i) = P(Q_t = S_i | O, M) = \frac{\alpha_t(S_i) \beta_t(S_i)}{P(O|M)}$$

Formel 2.30 A-posteriori-Wahrscheinlichkeit eines Zustands

Analog dazu besteht die A-posteriori-Wahrscheinlichkeit eines Zustandswechsels durch die genormte Wechselwahrscheinlichkeit des betrachteten Zeitpunkts, wie folgende Gleichung verdeutlicht: (vgl. Fink 2008, S. 81)

$$\gamma_t(S_i, S_j) = P(Q_t = S_i, Q_{t+1} = S_j | O, M) = \frac{\alpha_t(S_i) a_{i,j} b_j(O_{t+1}) \beta_{t+1}(S_j)}{P(O|M)}$$

Formel 2.31 A-posteriori-Wahrscheinlichkeit eines Zustandswechsels

Sobald die für diese Berechnungen benötigten Daten nach Abschluss des Forward-Backward-Algorithmus bereit liegen, beginnt der eigentliche Optimierungsschritt des Trainingslaufs. Dabei werden für jede Übergangswahrscheinlichkeit der Übergangsmatrix A , die prinzipiell möglich sind, die A-posteriori-Wahrscheinlichkeiten über alle Zeitintervalle 1 bis $T-1$ des Ursprungszustands $\gamma_t(S_i)$ und der denkbaren Zustandswechsel $\gamma_t(S_i, S_j)$ bestimmt. Aus diesen Werten ergibt sich dann der angepasste Eintrag der Übergangsmatrix $\hat{a}_{i,j}$ wie folgt: (vgl. Fink 2008, S. 82)

$$\hat{a}_{i,j} = \frac{\sum_{t=1}^{T-1} \gamma_t(S_i, S_j)}{\sum_{t=1}^{T-1} \gamma_t(S_i)}$$

Formel 2.32 Parameteranpassung mittels A-posteriori-Wahrscheinlichkeiten

Im Anschluss an die Anpassung der Parameter folgt solange ein weiterer Trainingslauf, bis die dabei erhobenen Produktionswahrscheinlichkeiten der beiden aktuellsten Modelle sich so wenig unterscheiden, dass der Anwender weitere Trainingsläufe für unnötig hält und das Verfahren terminiert. Der Baum-Welch-Algorithmus an sich bietet kein Abbruchkriterium. (vgl. Fink 2008, S. 77)

2.3 Hidden non-Markovian Models

HMM finden bereits Anwendung in der Erkennung von Mustern und Strukturen, wie der Spracherkennung. Jedoch stellt sich die Annahme, dass betrachtete System sei gedächtnislos, in vielen Fällen als unrealistisch aus. Hidden non-Markovian Models umgehen diese Einschränkung, indem sie statt einer DTMC ein Stochastisches Petrinetz als verborgendes Modell annehmen. (vgl. Krull 2008, S. 84) Diese verwenden Dichtefunktionen (pdf) von stetigen Wahrscheinlichkeits-Verteilungen, um den Zeitpunkt eines Zustandsübergangs abzubilden, womit die Übergangsraten auch zeitinhomogen sein können und zudem von dem dynamischen Verhalten des Modells in der Vergangenheit abhängen können. Damit ist die Markoveigenschaft des Modells nicht mehr gegeben. (vgl. Marsan 1989, S. 13)

Für die Simulation solcher Modelle bieten sich Discrete Event Simulation an, doch da die Ergebnisse einzelner Simulationsläufe nicht deterministisch sind und sich diese drastisch voneinander unterscheiden können, wurde eine mathematisch analytischer Ansatz entwickelt, mit dessen Hilfe in nur einem Simulationslauf ein deterministisches Ergebnis vorliegt. Hierbei handelt es sich um die auf Proxel basierende Simulation. (vgl. Horton 2002, S. 129–136)

Bei einem Proxel handelt es sich um einen Vektor $p = (\pi, Z)$, der aus einem Zustand $Z = (S_i, \tau, t)$ eines SPN und der Wahrscheinlichkeit π , dass sich das SPN im Zustand Z befindet. Der Zustand einer SPN setzt sich dabei aus der Markierung S_i , dem Zeitpunkt des Zustandes t und dem Vektor τ , der das Alter einer Aktivität, den Betrag an Zeit von aktiven Transitionen seit ihrer Freigabe bis zum Zeitpunkt t , auflistet. (vgl. Horton 2002, S. 132)

HnMM unterscheiden sich insofern von HMM, dass die Ausgabe von Symbolen, mit Hilfe derer auf das Verhalten des Modells geschlossen wird, nicht mehr in Abhängigkeit der Zustände des Modells geregelt sind, sondern von Zustandsänderungen hervorgerufen werden, die wiederum durch abgeschlossene Aktivitäten ausgelöst werden. Zudem müssen nicht alle Aktivitäten bei ihrem Abschluss ein Symbol senden. (vgl. Buchholz 2012, S. 15)

Dabei sind HnMM jedoch mit Problemen belastet. Abgesehen davon, ein kompletter Methodensatz fehlt, mit dem sich die Aufgaben der Evaluation, Decodierung und des Trainings lösen lässt, wächst bei den bereits vorhandenen Methoden der Aufwand exponentiell mit der Länge der Beobachtungssequenz. (vgl. Buchholz 2012, S. 15)

2.3.1 Spezifikation eines Conservative Hidden non-Markovian Model

Der allgemeine Ansatz der HnMM ist zwar mächtig, aber nicht effizient, weswegen dieser bereits weiterentwickelt wurde. Eine Variante, die durch Restriktionen eine Analyse des dynamischen Verhaltens eines HnMM in angemessener Zeit zu bewältigen gedenkt, ist die Klasse der Conservative Hidden non-Markovian Models (CHnMM). Diese gehen davon aus, dass alle abgeschlossenen Aktivitäten ein Symbol senden und dass der Abschluss keiner Aktivität die Zeitspanne bis zum Abschluss anderer Aktivität beeinflusst. (vgl. Buchholz 2012, S. 25–26)

Die CHnMMs übernehmen von den HMM den Zustandsraum $S = \{S_1, S_2, \dots, S_N\}$, wobei der einzelne Zustand eines Systems nicht mehr nur durch die Markierung $S_i \in S$ beschreibbar ist. Der Vektor τ muss zu diesem Zweck hinzugezogen werden. Das Verhalten des Modells $M = (A, B, \pi_0)$ über die Zeit kann jedoch wie gehabt über die Zustandssequenz $Q = Q_1, Q_2, \dots, Q_U \in S$ protokolliert werden kann, die nun aber nicht für jedes Zeitintervall ein Symbol enthält, sondern für jeden Zeitpunkt, in dem sich der Systemzustand geändert hat. Initiiert wird das Modell durch den Initialzustand π_0 , der in Form eines Vektors mit der Länge N die Wahrscheinlichkeitswerte enthält, mit der das Modell zum Zeitpunkt $t = 1$ sich in den jeweiligen Zuständen befindet. Ebenfalls

übernommen wird das Prinzip der von außen wahrnehmbare Symbolsequenz $O = O_1, O_2, \dots, O_U \in V$ mit der finiten Menge an möglichen Symbolen $V = \{V_1, V_2, \dots, V_L\}$ (vgl. Buchholz 2012, S. 28)

Die Übergangsmatrix A mit der Größe $N \times N$ wird hingegen angepasst. Sie enthält nicht mehr Werte für die schrittweise Übergangswahrscheinlichkeiten der Zustände untereinander, sondern komplexe Beschreibungen von Aktivitäten TR_i der SPN, die für Zustandsänderungen im Modell sorgen. Diese sind in der Menge $TR = \{TR_1, TR_2, \dots, TR_K\}$ zusammengefasst. Jedes Element dieser Menge besteht aus dem Tupel $(dist, id, b(v), aging)$. (vgl. Buchholz 2012, S. 28)

Der erste Eintrag $dist$ beschreibt die stetige Wahrscheinlichkeitsverteilung für die Dauer der Aktivität. Die $id \in \mathbb{N}$ ermöglicht es die Transition wiederzuerkennen und wird mit dem konstanten Wert $TR_i.id = i$ belegt. Die Funktion $b(O_u)$ berechnet auf Grundlage des erfassten Symbols O_u die Wahrscheinlichkeit, dass die betrachtete Tätigkeit im Falle des Abschlusses dieses Symbol ausgibt. Schlussendlich bestimmt $aging \in \{true, false\}$, ob eine Aktivität auch dann weiter fortschreitet ($aging = true$), wenn sie unterbrochen wurde, nachdem sie aktiviert wurde, oder ob das Alter in dem Fall unverändert bleibt ($aging = false$), bis die Aktivität wieder freigegeben wurde. (vgl. Buchholz 2012, S. 28–29)

Ein Element der Übergangsmatrix A besteht demnach aus der einen Aktivität, die für die betrachtete Zustandsänderung verantwortlich ist, oder aus einem Symbol für die Leere, falls der betreffende Zustandswechsel nicht möglich ist. Zusätzlich kann nur eine Aktivität nur einmal pro betrachteten Zustand ausgelöst werden und hat daher ein eindeutiges Ziel in Form eines anderen Zustandes. (vgl. Buchholz 2012, S. 29)

Außerdem wird dem SPN des versteckten Modells hinter der CHnMM die Restriktion auferlegt, dass zwei zur gleichen Zeit aktive Transitionen nie die gleiche Markierung setzen können, und dass von den Transitionen, die sofort feuern, nur eine zu gleich aktiv sein kann. Die Transitionen, die erst nach einer gewissen Zeit im aktivierten Zustand abgeschlossen werden, senden außerdem Symbole. Dieser Sachverhalt wird durch gestrichelte Pfeile von der Transition zum Symbol, versehen mit einem Wahrscheinlichkeitswert, verdeutlicht. Aufgrund dieser Ergänzung wurde diese Art der SPN Augmentend Stochastic Petri Net (ASPN) getauft. (vgl. Buchholz 2012, S. 20–21)

2.3.2 Anwendungsfall Evaluation

Für die Berechnung der Produktionswahrscheinlichkeit $P(O|M)$ werden lediglich die Zeitpunkte des der Symbol der Sequenz O durch das Modell M betrachtet. Daher bedarf es der Bestimmung der *Verweilwahrscheinlichkeit* $P_{sojourn}$, mit der ein Modell im Zustand Z des Proxel $p = (\alpha, Z) = (\alpha, (S_i, \tau, t_{old}))$ vom Zeitpunkt t_{old} bis zum Zeitpunkt t_{new} verbleibt. Für diesen Zweck werden alle Aktivitäten $\{TR_1, TR_2, \dots, TR_k\}$ betrachtet, die eine Zustandsänderung von S_i aus erwirken können. Demnach gilt: (vgl. Buchholz 2012, S. 40)

$$P_{sojourn}(p) = P(\forall 1 \leq j \leq n : TR_j > t_{new} | \forall 1 \leq j \leq n : TR_j > t_{old})$$

Formel 2.33 Psojourn

In diesem Zusammenhang bedeuten $TR_i > t$, dass TR_i zu einem späteren Zeitpunkt als t abgeschlossen wird. Berechnet wird die Verweilwahrscheinlichkeit anhand des Produkts aller kumulativen Verteilungsfunktionen (CDF) der Aktivitäten, nach der folgenden Gleichung: (vgl. Buchholz 2012, S. 41)

$$P_{sojourn}(p, t_{new}) = \prod_{j=1}^n \frac{P(TR_j > t_{new})}{P(TR_j > p \cdot t_{old})} = \prod_{j=1}^n \frac{1 - F_j(\tau_j + (t_{new} - p \cdot t_{old}))}{1 - F_j(\tau_j)}$$

Formel 2.34 Psojourn Berechnung

Zudem wird die Wahrscheinlichkeit P_{change} betrachtet, dass Aktivität TR_j genau im betrachteten Augenblick t_{new} abgeschlossen wird. Diese ist eigentlich verschwindend gering. Um dennoch mit P_{change} arbeiten zu können, wird dieser gegen Null strebenden Größe einen Wert zugewiesen, bekannt als die virtuelle Wahrscheinlichkeit, die eine entscheidende Eigenschaft mit dem realen Wert von P_{change} teilt: Die Verhältnisse aller realen Werte von P_{change} untereinander entsprechen ihren virtuellen Gegenstücken. Da bei der Evaluation vor allem der Vergleich von verschiedenen Modellen untereinander zählt, reicht diese Eigenschaft für die weiteren Berechnungen aus. Zu beachten ist jedoch, dass durch CHnMMs berechnete Produktionswahrscheinlichkeit nicht mehr der Realität entspricht. (vgl. Buchholz 2012, S. 41–45)

Der virtuellen Wahrscheinlichkeit P_{change} wird der Wert der instantaneous rate function $\mu_j(t)$ der aktuellen Verteilung zugewiesen (vgl. Buchholz 2012, S. 44) und wird wie folgt berechnet: (vgl. Horton 2002, S. 232)

$$P_{change}(TR_j, t) = \mu(TR_j, t) = \frac{TR_j \cdot dist. f(t)}{1 - TR_j \cdot dist. F(t)}$$

Formel 2.35 Pchange

Evaluiert wird ein Modell mit einen auf CHnMMs angepassten Forward-Algorithmus, der für jedes beobachtetes Symbol $O_u \in O$ zu deren Zeitpunkt t eine Menge R_u an Proxel erstellt, auf deren Grundlage wiederum die Proxelmenge R_{u+1} folgt. Der Wert $p.\alpha_u$ entspricht dabei im Grunde der Forward-Wahrscheinlichkeit aus Abschnitt 2.2.3, ist aber virtuell. Initialisiert wird dieser Prozess zum Zeitpunkt 0 anhand von π_0 und mit einem Nullvektor mit jeweils einen Eintrag für alle Aktivitäten von S_i , wobei wie die nächste Gleichung zeigt: (vgl. Buchholz 2012, S. 46)

$$R_0 = \bigcup_{S_i \in \{S_1, S_2, \dots, S_N | \pi_0(S_i) \neq 0\}} \{(\pi_0(S_i), (S_i, \mathbf{0}, 0))\}$$

Formel 2.36 Initialzustand der CHnMM

In den nachfolgenden Schritten u wird dann zum Füllen der Menge R_u jeder einzelne Proxel p aus der Menge R_{u-1} betrachtet. Für jedes p wird erst die Verweilwahrscheinlichkeit $P_{\text{sojourn}}(p, t)$ bestimmt und dann einzeln für jede Aktivität TR_j der Markierung S_i von p die Wahrscheinlichkeit $P_{\text{change}}(TR_j, t)$ berechnet. Die Multiplikation von $P_{\text{sojourn}}(p, t)$ und $P_{\text{change}}(TR_j, t)$ ergibt wiederum die Übergangswahrscheinlichkeit $\text{transProp}(p, TR_j, t)$. Schließlich ergibt sich aus den so gewonnen Werten ein neuer Proxel der Menge R_u , der Markierung S_j , zu der TR_j von S_i aus führt, wobei im Zuge dessen die Elemente von τ entsprechend der vergangenen Zeit zwischen den Beobachtungen O_{u-1} und O_u anpassen, woraus τ' folgt. Damit gilt folgendes: (vgl. Buchholz 2012, S. 46)

$$R_u = R_{u-1} \cup \{(p.\alpha * \text{transProp}(p, TR_j, t) * TR_j.b(O_u), S_j, \tau', t)\}$$

Formel 2.37 Folgezustände der CHnMM

Folgt man starr diesem Vorgehen, vermehrt sich die Anzahl der zu verarbeitenden Proxel exponentiell mit der Länge der Beobachtungssequenz. Dabei muss jedoch nicht jeder aus dieser Gleichung resultierender Proxel tatsächlich in R_u aufgenommen werden. Proxel, der Wahrscheinlichkeit Null entspricht, brauchen nicht weiter betrachtete werden. Dies ist vor allem der Fall wenn das Symbol O_u von der gerade betrachteten Aktivität nicht ausgelöst werden konnte. Ist zudem in der Menge R_u bereits ein Proxel vorhanden, dessen Modellzustand $p.Z = (S_j, \tau, t)$ dem des aufzunehmenden Proxel p' entspricht, muss dessen Forward-Wahrscheinlichkeit lediglich auf $p.\alpha$ addiert werden, womit die beiden Proxel verschmelzen. Dies hilft dabei, die Menge an zu verarbeitenden Proxel möglichst gering zu halten. Die virtuelle Produktionswahrscheinlichkeit $P(O|M)$ ergibt sich dann aus der Summe der Forward-Wahrscheinlichkeit aller Proxel von R_u , gemäß der Formel 2.38 (vgl. Buchholz 2012, S. 46–49)

$$P(O|M) = \sum_{p \in R_u} p \cdot a$$

Formel 2.38 CHnMM Evaluation Ergebnis (vgl. Buchholz 2012, S. 47)

2.3.3 Anwendungsfall Decodierung

Ähnlich wie in Abschnitt 2.2.4 wird die Sequenz an Markierungen Q von Modell M gesucht, die angesichts der beobachteten Symbolsequenz O das Modellverhalten am ehesten erklärt, wie Formel 2.39 Formel 2.39 CHnMM Decodierung zeigt:

$$Q^* = \operatorname{argmax}_Q P(Q|O, M)$$

Formel 2.39 CHnMM Decodierung

Auch in diesem Fall hat das naive Vorgehen, die Wahrscheinlichkeit jeder möglichen Sequenz Q zu berechnen und diejenige mit der höchsten auszuwählen, einen Aufwand zur Folge, der mit der Länge der Beobachtungssequenz exponentiell wächst. Wenn jedoch beim Forward-Algorithmus aus Abschnitt 2.3.2 Proxel, die identische Modellzustände repräsentieren, nicht verschmolzen werden, sondern durch Proxel mit dem höheren Wert der Forward-Wahrscheinlichkeit ersetzt werden, steht hinter jeder Forward-Wahrscheinlichkeit eines überlebenden Proxel die virtuelle Pfadwahrscheinlichkeit, deren Sequenz an Modellzuständen zudem zurückverfolgt werden kann, sofern jeder Proxel mit einem Link zu seinem einzigen Erzeuger verbunden ist. (vgl. Buchholz 2012, S. 60–62)

Q^* ergibt sich in diesem Falle aus dem Proxel mit der höchsten Forward-Wahrscheinlichkeit von R_U , siehe folgende Gleichung. (vgl. Buchholz 2012, S. 60–62)

$$Q^* = \operatorname{argmax}_{p \in R_u} p \cdot a$$

Formel 2.40 CHnMM Decodierung Pfadauswahl (vgl. Buchholz 2012, S. 60–62)

2.3.4 Anwendungsfall Training

Wie auch beim Training eines HMM (siehe Abschnitt 2.2.5) werden auf der Suche nach ein Modell M' , dessen Produktionswahrscheinlichkeit $P(O|M')$ für die Symbolsequenz O besser oder wenigsten genauso gut ist wie $P(O|M)$, die Parameter von Modell M auf dem Weg zu einem lokal optimalen $P(O|M')$ wiederholt schrittweise angepasst. (vgl. Buchholz 2012, S. 89)

Im Falle der CHnMM existieren, neben der Möglichkeiten zur Anpassung von Initialzustand und der Wahrscheinlichkeiten der Aktivität für das Senden des jeweiligen

Symbols bei Abschluss, Mittel sowohl den Erwartungswert wie auch die Standardabweichung der Verteilung einer Aktivität zu schätzen. Es bietet sich demnach am ehesten bei Verteilungen an, deren Parameter sich durch diese beiden Größen bestimmen lassen. (vgl. Buchholz 2012, S. 98)

Der erste Schritt zu einer Schätzung besteht wie bei der HMM in der Anwendung des Forward-Algorithmus aus Abschnitt 2.3.2, mit den kleinen aber bedeutenden Unterschied, dass für alle Proxel deren Erzeugerproxel rekonstruierbar sein und dementsprechend im Speicher der ausführenden Rechenmaschine abgelegt werden müssen. Dies führt zu einen noch größeren Speicherplatzverbrauch als bei der Decodierung, da wie bei der Evaluation die Forward-Wahrscheinlichkeiten von Proxel mit identischen Zustand Z aufeinander addiert werden und sie somit ihre Erzeuger teilen, statt alle bis auf einen Pfad Q zu entfernen. (vgl. Buchholz 2012, S. 97)

Alle auftretenden Zustandssequenzen in Form von aufeinander folgenden Proxel zu erhalten ist jedoch notwendig, da auch beim Training der CHnMM die Backward-Wahrscheinlichkeiten eine Rolle spielen. Demnach enthält ein Proxel p der Menge R_u beim Training den Wert Forward- und der Backward-Wahrscheinlichkeit, den Modellzustand Z und ein Vektor von Erzeugerproxel aus der Menge R_{u-1} . (vgl. Buchholz 2012, S. 97)

Dafür bedarf es wie bei der HMM die Berechnung der Backward-Wahrscheinlichkeit, wofür der zweite Schritt des hier verwendeten Forward-Backward-Algorithmus dient. Die Backward-Wahrscheinlichkeit eines jeden Proxel p der Menge R_U entspricht Eins und für jede Verbindung von jeden Proxel p' in R_u zu dessen Erzeuger p in R_{u-1} wird eine Backward-Wahrscheinlichkeit errechnet, gemäß der folgenden Formel. Daher setzt sich der Gesamtwert von $p.\beta$ aus der Summe der Backward-Wahrscheinlichkeiten der Verbindungen zu allen erzeugten Proxel zusammen. (vgl. Buchholz 2012, S. 81)

$$p.\beta += \text{transProp}(p, TR_j, t) * p'.\beta$$

Formel 2.41 CHnMM Backward-Wahrscheinlichkeit (vgl. Buchholz 2012, S. 81)

Für die Schätzung von Erwartungswerten und Standardabweichungen der verwendeten Wahrscheinlichkeitsverteilungen der Aktivitäten TR des betrachteten Modells ist dann vor allem die Wahrscheinlichkeit $\text{pathProb}(p, p')$ einer jeden Verbindung durch die Aktivität TR_j zwischen einen Proxel p' und dessen Erzeuger p wichtig. Diese setzt sich wie folgt zusammen: (vgl. Buchholz 2012, S. 81)

$$\text{pathProb}(p, p') = p.\alpha * \text{transProb}(p, p'.TR_j, p'.t) * p'.\beta$$

Formel 2.42 CHnMM pathProb (vgl. Buchholz 2012, S. 81)

Dieser Wert fließt entweder direkt oder multipliziert mit der Dauer der Aktivität in die Schätzungen ein. Dabei fällt ein geschätzter Erwartungswert einer Aktivität umso höher aus, je höher die relative Summe der pathProb der beteiligten Verbindungen ausfällt und je größer die Dauer der Aktivität ist.

3 FALLSTUDIE EINER SYSTEM RELIABILITY ANALYSIS

Das Ziel dieser Fallstudie ist die exemplarische Anwendung eines Hidden non-Markovian Model im Rahmen einer Reliability Analysis an einem realen Beispiel. Konkret handelt es sich bei dem untersuchten System um eine automatisierte Produktionslinie, die tiefgekühlte Pizza herstellt. Dieses Beispiel wurde willkürlich gewählt. Alle für die Anwendung der Zuverlässigkeitsanalyse relevanten Informationen stammen aus einer Arbeit, die unter anderem zu eben diesen Zweck geschrieben wurde, anderen Analysten es zu ermöglichen, sich an einer Zuverlässigkeitsanalyse zu versuchen, ohne selbst Daten erheben müssen. (vgl. Liberopoulos 2003, S. 1)

Das Paper stellt jedoch nur die Exponential- und Weibullverteilungen der betrachteten Produktionsmaschinen, externer Versorgungssysteme und der wichtigsten Fehlerquellen den Analysten. Das Protokoll der Beobachtungen, das über einen Zeitraum von vier Jahren und einen Monat angefertigt wurde, ist nicht Teil der Arbeit über die Pizza-Produktionslinie. (vgl. Liberopoulos 2003, S. 6–18) Daher wurde diese Produktionslinie anhand der Beschreibungen und Daten aus dem Paper im Simulator AnyLogic® nachgebaut. Die Sequenz an beobachteten Symbolen wurde durch das Simulationsmodell erzeugt, das dem digitalen Anhang dieser Arbeit beiliegt. Sie enthält den jeweiligen Zeitpunkt und das ausgebende Symbol einer jeden Zustandsänderung innerhalb des Beobachtungszeitraum, wobei beim Symbol zuerst die Arbeitsstation („WS“), dann die Maschine („M“) identifiziert wird. Dann folgt entweder ein I für einen kritischen Fehler oder A für eine abgeschlossene Reparatur. Die Symbolsequenz ist fast 3200 Einträge lang.

3.1 Ausgangslage und Ziele der Studie

Der genaue Beobachtungszeitraum, in dem die Daten über die Produktionslinie erhoben wurden, beträgt 883 Tage mit jeweils drei Schichten. Die Arbeitszeit pro Schicht betrug 8 Stunden. Aufgenommen wurden dabei für alle betrachteten Systemkomponenten die Anzahl der Schichten time-between-failure(tbf) zwischen zwei Ausfällen derselben Maschine und die Zeit in Minuten, die für ihre Reparatur benötigt wurde. (vgl. Liberopoulos 2003, S. 6)

Die Produktionslinie besteht dabei aus 14 Maschinen, unterteilt in 6 Arbeitsstationen, die nacheinander die Arbeitsschritte an den Produkten ausführen. In der Arbeitsstationen 7 sind die Versorgungssysteme für elektrische Energie, Wasser, Gas und Luft

zusammengefasst. Ein kritischer Ausfall in eines dieser Systeme hat einen Ausfall der gesamten Produktionslinie zur Folge. (vgl. Liberopoulos 2003, S. 5–6)

In der ersten Arbeitsstation wird der Teig geknetet. Bei der ersten Maschine handelt es sich um ein Silo, von dem Mehl aus automatisch zusammen mit Wasser in eine Schale gefüllt wird, die in der zweiten Maschine, einer Knetmaschine, hängt. Dort wird zudem manuell kleinere Menge anderer Zutaten dem Teig beigemischt. Bei der dritten Maschine der ersten Arbeitsstation handelt es sich um eine Hebevorrichtung, die manuell mit der Schale aus der Knetmaschine beladen wird. Sie kippt den Teig in den Extruder der zweiten Arbeitsstation. (vgl. Liberopoulos 2003, S. 4–6)

Diese ist für die Laminierung des Teigs und dem anschließenden Ausschneiden der Formen zuständig. Die erste Maschine dieser Arbeitsstation walzt den Teig auf eine bestimmte Dicke, die zweite trennt die Formen heraus. Die Pizzaböden gelangen anschließend auf Backbleche, die an die dritte Arbeitsstation weitergegeben werden. (vgl. Liberopoulos 2003, S. 4–6)

Hier wird in einer einzigen Maschine der Belag auf die Pizzen gestreut. Um zu verhindern, dass der Teig während des Backens hochgeht, geschieht dieser Schritt davor. (vgl. Liberopoulos 2003, S. 4–6)

Anschließend kommen die belegten Pizzen in den vorgeheizten Ofen, der einzigen Maschine der vierten Arbeitsstation. In diesem verweilen die Pizzen eine bestimmte Zeit lang. (vgl. Liberopoulos 2003, S. 4–6)

In der fünften Arbeitsstation ruht der Teig. Zu diesem Zweck werden die Backbleche mit den Pizzen in der ersten Maschine gesammelt, dann von der zweiten zu ihren Ruheplätzen transportiert, an denen die Pizzen abkühlen, was die dritte Maschine darstellt. Dank der vierten Maschine werden die Backbleche mitsamt der Pizzen wieder aus der Arbeitsstation entfernt. (vgl. Liberopoulos 2003, S. 4–6)

In der sechsten Arbeitsstation werden die Pizzen durch die erste Arbeitsmaschine von den Blechen gehoben, durch die zweite Maschine werden sie versiegelt und auf die dritte Maschine, einem Förderband geladen. Von dort aus werden die Pizzen manuell in Kartons verpackt und abtransportiert.

Da es sich hier um ein Seriensystem handelt, verursacht der Ausfall einer Maschine den Stop der gesamten Produktionsanlage stromaufwärts von der ersten Maschine der ersten Arbeitsstation bis zur ausgefallenen. Stromabwärts der funktionsuntüchtigen Maschine

kann zwar noch weiter produziert werden, dennoch kommt es zu einer totalen Unterbrechung der Produktion für den Zeitraum, bis die ausgefallene Maschine wieder repariert ist, da neue Produkte nicht nachkommen. Pizzen, die länger als 25 Minuten in stromaufwärts unbewegt verharren, müssen zur Sicherung der Produktqualität entsorgt werden, wodurch die Produktionslücke in Abhängigkeit der Dauer wächst, die eine Pizza bis zur kaputten Maschine braucht. Fällt der Ofen direkt aus, kommt es zudem zu einem Temperaturabfall. Dauert die Reparatur des Ofens mehr als 5 Minuten, muss dieser erst wieder vorgeheizt werden, bevor die Produktion wieder aufgenommen werden kann. All diese Gegebenheiten wurden in der Arbeit zur Pizza-Produktionslinie in Regeln abgebildet, die wiederum im Zuge dieser Arbeit entwickelten Simulationsmodells berücksichtigt wurden. (vgl. Liberopoulos 2003, S. 6–7)

Die gesamte Pizzaproduktion lässt sich beobachten. Ein verstecktes Verhalten, dass die Anwendung eines HMM oder HnMM gibt es diesem Sinne nicht. Doch dies ist absolut im Sinne des Autors, da so die Ergebnisse dieser beiden Modelle überprüfbar werden. Daher wird ein versteckter Aspekt der Produktion angenommen.

Im Rahmen dieser Arbeit besteht die aufgenommene Beobachtungssequenz daher aus Symbolen die jeweils für einen Ausfall und der Reparatur einer Maschine aus einer bestimmten Arbeitsstation. Somit ist Zeitpunkt und die betreffende Arbeitsstation eines beobachteten Ereignis bekannt, aber nicht welche Maschine der Arbeitsstation gerade ausgefallen ist oder repariert wurde.

Zwar hätte sich an dieser Stelle auch angeboten, die ausgefallene oder wieder instand gesetzte Maschine genau identifizieren zu können und stattdessen die Fehlerquelle des kritischen Ausfalls als unbekannt anzusehen, doch das Paper zur Pizzaproduktion gab für diesen Fall nicht genug Daten her, da nur die wichtigsten Fehlerquellen dokumentiert wurden. Für einige der Maschinen ist keine Fehlerquelle verzeichnet. (vgl. Liberopoulos 2003, S. 11–12)

Da die Beobachtungssequenz aus einem Simulationsmodell stammt, handelt es sich um Daten von höchster Qualität. Der Zeitpunkt eines Ausfalls und der Zeitpunkt des Abschlusses einer Instandsetzung sind in Minuten verzeichnet und auf zehn Stellen hinter dem Komma genau protokolliert.

Ziel der Studie ist es aus diesen Daten das wahrscheinlichste Verhalten der Produktionslinie angesichts zu rekonstruieren und die Parameter des Modells zur

Abbildung der Ausfälle und der Instandsetzung von Maschinen angesichts der beobachteten Ereignisse anzupassen.

Arbeitsstation.Maschine	tbf mean	tbf weibull scale	tbf weibull shape
1.1	31.8289	23.5973	0.6227
1.2	49.4082	49.6005	1.0059
2.1	10.0471	9.5125	0.6949
2.2	7.2297	5.9125	0.6995
3.1	12.8469	10.9148	0.6776
4.1	26.0693	22.3624	0.7532
5.1	22.9821	22.4771	0.6955
5.2	30.9294	27.6181	0.6863
5.3	25.3960	20.5398	0.5513
5.4	35.2394	30.3546	0.7525
6.1	19.0292	17.1183	0.7755
6.2	26.6768	24.0122	0.6911
6.3	40.8226	29.6937	0.8202
7.1	49.8800	28.5554	0.6085

Tabelle 3.1 time between failure der Maschinen in 8-Stunden-Schichten (vgl. Liberopoulos 2003, S. 8–18)

Allerdings fiel ein paar der Maschinen im Beobachtungszeitraum des Papers zur Pizzaproduktion entweder gar nicht oder nicht oft genug aus um Ausfallrate und Reparaturrate bestimmen zu können. Dies betrifft die dritte Maschine der ersten Arbeitsstation, sowie bei der Arbeitsstation sieben die Versorgung mit Wasser, Gas und Luft. Diese werden bei der Zuverlässigkeitsanalyse im Rahmen dieser Arbeit nicht betrachtet. Damit verbleiben 14 Maschinen, die im Modell der Produktionslinie abgebildet werden. (vgl. Liberopoulos 2003, S. 9)

Da wie in Abschnitt 2.1.2 bereits erwähnt die Ausfallrate einer Maschine nicht zeithomogen ist, ist es wenig realistisch für die Zeit zwischen den Ausfällen und die Dauer der Instandsetzungsmaßnahmen exponentielle Verteilungen anzunehmen. Tatsächlich

stellt die Arbeit zu Pizzaproduktion neben den Parametern der Exponentialverteilungen auch die Parameter der entsprechenden Weibullverteilungen zur Verfügung. Diese werden zudem als passender bezeichnet. Daher bietet sich rein von der Datengrundlage der Studie her die Verwendung einer HnMM an. (vgl. Liberopoulos 2003, S. 16–18)

Arbeitsstation.Maschine	ttr mean	ttr weibull scale	ttr weibull shape
1.1	47.7922	52.0758	1.8555
1.2	40.3000	45.2911	2.7082
2.1	47.9492	53.4389	3.0311
2.2	39.7905	44.6892	2.2222
3.1	21.9036	24.4637	2.2536
4.1	31.2255	34.0459	1.5630
5.1	20.7965	23.0723	2.9601
5.2	37.9651	42.2034	1.9697
5.3	29.9020	33.1680	2.9783
5.4	21.2500	23.3516	3.7846
6.1	27.5362	30.6924	2.2242
6.2	37.0000	41.4618	1.9753
6.3	12.9365	14.2368	3.5502
7.1	42.9412	48.2671	1.8481

Tabelle 3.2 time to repair der Maschinen in Minuten (vgl. Liberopoulos 2003, S. 8–18)

Tabelle 1 listet die durchschnittliche Anzahl an Arbeitsschichten zwischen den Ausfällen (tbf mean) für jede Maschine auf und führt für die für die Weibullverteilungen, die jeweils das Ausfallverhalten einer Maschine abbilden, die Form- (tbf weibull shape) und Skalenparameter (tbf weibull scale) auf. Tabelle 2 bietet dieselben Information für die Reparaturen der Maschinen in Minuten.

Von den Aktivitäten dieser HnMM setzt keine eine andere zurück. Da zudem davon auszugehen ist, dass alle Vorkommnisse protokolliert wurden, müssen Zustandsänderungen abseits der Beobachtungen nicht verfolgt werden. Sofern es also

gelingt, ein ASPN des Ausfallverhaltens der Produktionslinie zu erstellen, das alle Zustandsänderungen mit Symbolbeobachtungen belegt, bietet sich die Verwendung der Klasse der Conversive Hidden non-Markovian Model an.

3.2 HMM der Produktionslinie

Zwar liegt der Fokus dieser Arbeit auf HnMMs als Mittel der Wahl, doch wird zum Zweck einer Benchmark ebenfalls ein HMM der Pizzaproduktionslinie vorgestellt. Da bei der Entwicklung der CHnMM auf Teile des HMM zurückgegriffen wird, folgt der Abschnitt zur Entwicklung des CHnMM dem der HMM.

3.2.1 Struktur des Modells

Das größte Hindernis bei der Entwicklung eines Modells der betrachteten Produktionslinie auf der Grundlage eines Zustandsraums ist dessen Umfang. Im Falle einer Zuverlässigkeitsanalyse, bei der jede Systemkomponente zwei unterschiedliche Zustände annehmen kann, funktionstüchtig und funktionsuntüchtig, muss jede mögliche Kombination der Zustände aller Maschinen berücksichtigt werden. Damit hätte ein HMM der ganzen Produktionslinie $2^{14} = 16384$ Zustände. Die Übergangsmatrix A würde dann genau 268.435.456 über Einträge verfügen. Ein solcher Umfang des Zustandsraums ist ganz klar unangemessen, vor allem wenn man bedenkt, dass es sich bei einem Ausfall von einer Maschine sowieso um ein seltenes Ereignis handelt. Dass zwei solcher Ausfälle sich überschneiden, kommt sogar noch seltener vor. Bei der durch das Simulationsmodell erzeugten Spur überlappten sich nur 2,4% der Ausfälle mit anderen. Daher bietet es sich eher an, jede Maschine einzeln zu betrachten, da diese kaum von dem Ausfallverhalten anderer Maschinen beeinflusst werden. Das Modell der Produktionslinie besteht daher aus der sortierten Menge $M = \{M_1, M_2, \dots, M_{14}\}$, wobei M_1 der ersten Maschine in der Produktionsreihe entspricht und M_{14} der letzten.

Das HMM einer Maschine hat dann genau zwei Zustände wie in Abschnitt 2.1.4. In dem einen Zustand ist die Maschine funktionstüchtig, im anderen Zustand ist sie es nicht. Dabei gilt es jedoch zwei Dinge zu bedenken. Zum einen wird im Rahmen dieser Arbeit davon ausgegangen, dass eine Maschine nicht ausfallen kann, solange sie wegen des Defekts einer anderen Maschine stromabwärts deaktiviert wurde. Es ist jedoch unerheblich welche Maschine ausgefallen ist.

Somit lässt sich die Verfügbarkeit einer Maschine über eine Kombination aus zwei Subsystemen darstellen. Ein System repräsentiert die aktuell betrachtete Maschine, das andere System bildet das Ausfallverhalten der Produktionslinie stromabwärts nach. Das daraus resultierende Modell M_i hat vier Zustände. Der Zustandsraum folgt dem bereits vorgestellten Modell in Abschnitt 2.1.4. Folglich bezeichnet $S_1 = AA$ den Zustand, dass sowohl die betrachtete Maschine wie auch sämtliche Maschinen stromabwärts funktionstüchtig sind. Im Falle von $S_2 = AI$ ist mindestens eine Maschine stromabwärts ausgefallen, die Maschine im Fokus des Modells ist jedoch betriebsbereit, wenn auch inaktiv. $S_3 = IA$ steht dann für den Umstand, dass sowohl mindestens eine Maschine stromabwärts, wie auch die betrachtete Maschine defekt sind. Schließlich weist bei S_4 die aktuelle Maschine einen kritischen Fehler auf, die Maschinen stromabwärts sind jedoch in Ordnung.

Dabei bezeichnet $tbf1_i$ den Kehrwert des Erwartungswerts der Zeit zwischen zwei Ausfällen der betrachteten Maschine, gemäß den Werten aus Tabelle 1, und ist damit äquivalent zur Ausfallrate. Die Reparaturrate $ttr1_i$ entspricht analog dazu dem Kehrwert der durchschnittlichen Dauer der Reparatur der betrachteten Maschine aus Tabelle 2. Beide Raten sind zeithomogen. Die Ausfallrate $tbf2_i(t)$ und Reparaturrate $ttr2_i(t)$ hingegen sind zeitinhomogen, da sie das dynamische Verhalten der Verfügbarkeit der Maschinen stromabwärts abbilden. Das versteckte Modell hinter der HMM in Form der DTMC sieht dann wie folgt aus:

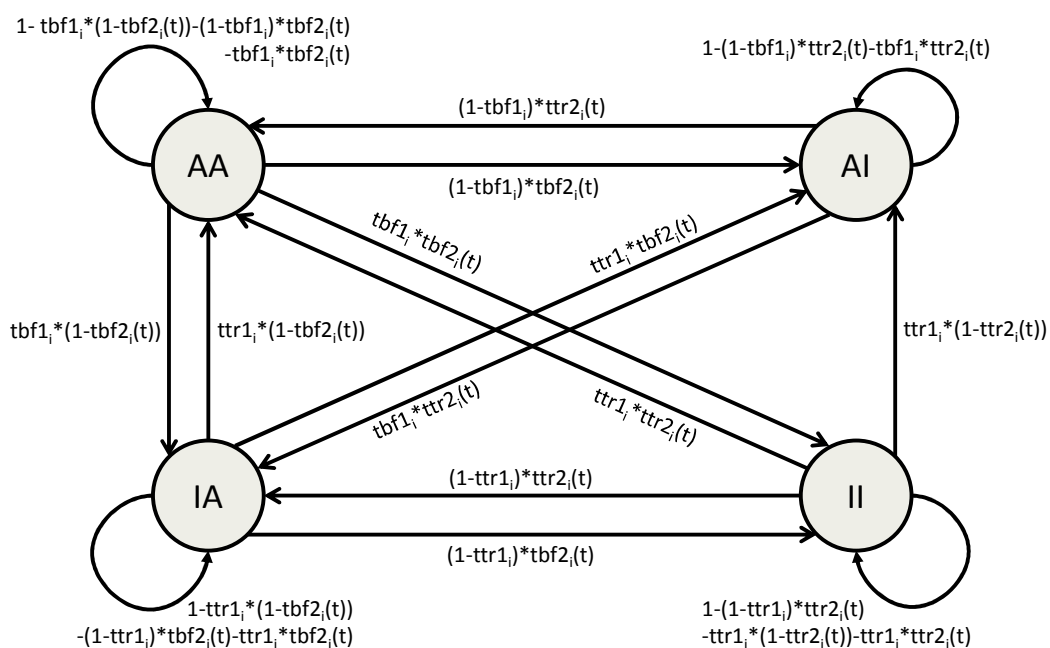


Abbildung 3.1 DTMC einer Maschine von M

Alle Werte der Übergangsmatrix müssen A in ein und derselben Einheit verzeichnet sein, die zugleich die Größe der Zeitintervalle festlegt. Um nach Möglichkeit kein Ereignis zu verpassen, wird im Rahmen dieser Arbeit die Intervalllänge von einer Minute festgelegt. Da der Beobachtungszeitraum insgesamt 883 Tage beträgt, werden für die Anwendungsfälle der HMM 1271520 Intervalle betrachtet.

Die Einträge von A bestehen stets aus einem Produkt, da das Modell zwei Subsysteme abbildet, deren Verhalten einen Seriensystems entspricht (siehe Formel 2.6). Die diagonalen Kanten sind nötig, da eine DTMC Intervalle betrachtet, in denen auch mehrere Maschinen ihren Zustand ändern können. Lediglich die Kante von AI nach II wurde weggelassen, da eine Maschine keinen kritischen Ausfall erleiden kann, solange mindestens eine Maschine stromaufwärts defekt ist. Es müsste schon innerhalb einer Minute die defekte Maschine stromaufwärts repariert werden, die betrachtete Maschine i müsste ausfallen und wiederum eine Maschine stromabwärts kaputt gehen, damit diese ausgelassene Kante genutzt werden würde. Sollte ein solcher Fall auftreten, sollte über eine Anpassung des Intervalls nachgedacht werden.

Ein Sonderfall in der Menge M stellt das Subsystem M_{14} dar. Dieses repräsentiert die Stromversorgung der Produktionslinie. Ob dieses System ausfallen kann oder nicht, ist von keinem anderen System abhängig. Daher entspricht die Struktur des Modells M_{14} der Abbildung 2.6, wobei λ der Rate $tbfl_{14}$ von entspricht und μ der Rate $ttr1_{14}$. Den Umstand, dass ein Ausfall der Stromversorgung den Stillstand der gesamten Produktionslinie zur Folge hat, wird Rechnung getragen, indem die Stromversorgung als Maschine ganz am Ende stromabwärts der Produktionslinie platziert wird. So schaltet sie alle anderen Maschinen mit aus, sollte sie defekt sein.

Die $tbf2_{13}(t)$ der Maschine M_{13} entspricht somit der $tbfl_{14}$ vom M_{14} . Analog gilt dies für die $ttr2_{13}(t)$ von M_{13} und $ttr1_{14}$ von M_{14} . Diesen Gedanken folgt das iterative Verfahren, mit Hilfe dessen auch die $tbf2_i$ - und $ttr2_i$ - Raten der anderen Maschinen errechnet werden können. Wenn ein Modell M_{i+1} mit den vier Zuständen $S_1 = AA$, $S_2 = AI$, $S_3 = II$ und $S_4 = IA$ auf ein Modell W mit den Zuständen $W_1 = S_1$ und $W_2 = S_2 \cup S_3 \cup S_4$ und einer Übergangsmatrix nach Formel 3.1 übertragen wird, lässt aus der $tbfl_w(t)$ und $ttr1_w(t)$ dieses Modells auf die $tbf2_i$ und $ttr2_i$ von M_i schließen.

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{pmatrix} = \begin{pmatrix} 1 - tbf1 & tbf1 \\ ttr1 & 1 - ttr1 \end{pmatrix}$$

Formel 3.1 Übergangsmatrix einer einzelnen Maschine

Um dies zu erreichen, werden die Wahrscheinlichkeit $\pi_t(W_i)$ zum Zeitpunkt t und $\pi_{t+1}(W_i)$ miteinander verglichen. Für diese gilt:

$$\pi_t(W_1) = \pi_t(S_1)$$

$$\pi_t(W_2) = \pi_t(S_2) + \pi_t(S_3) + \pi_t(S_4)$$

Formel 3.2 Modellüberführung

Daraus folgt in Anlehnung an Formel 2.13:

$$\pi_{t+1}(W_1) = \pi_t(W_1)(1 - tbf_w(t)) + \pi_t(W_2)ttr_w(t)$$

$$\pi_{t+1}(W_2) = \pi_t(W_2)(1 - ttr_w(t)) + \pi_t(W_1)tbf_w(t)$$

Formel 3.3 Gleichungssystem zur Bestimmung von Ausfall- und Reparaturrate

Leider lässt sich dieses Gleichungssystem nicht direkt lösen, da es unendlich viele Lösungen hat. Jedoch lässt sich die Rate $tbf_w(t)$ direkt ablesen, denn hierbei handelt es sich lediglich um die aufsummierten Gewichte der Kanten, die von S_1 aus zu den anderen Zustände gehen, die allesamt in W_2 liegen (siehe Formel 3.4). Da $tbf_w(t)$ von W bezüglich dem Modell M_i nun bekannt ist, lässt sich aus dem Gleichungssystem Formel 3.3 auch die Reparaturrate $ttr_w(t)$ bestimmen.

$$\begin{aligned} tbf_w(t) &= tbf_{1_{i+1}}(1 - tbf_{2_{i+1}}(t)) + tbf_{2_{i+1}}(t)(1 - tbf_{1_{i+1}}) + tbf_{1_{i+1}}tbf_{2_{i+1}}(t) \\ &= tbf_{1_{i+1}} + tbf_{2_{i+1}} - tbf_{1_{i+1}}tbf_{2_{i+1}}(t) = tbf_{2_i}(t) \end{aligned}$$

Formel 3.4 Dynamische Ausfallrate tbf_2

Wie bereits erwähnt, sind die Ausfall- und Reparaturrate der Maschinen stromabwärts nicht zeithomogen. Dies beeinflusst jedoch nicht im Geringsten die Markoveigenschaft der Gedächtnislosigkeit des Modells, da sich die zeitinhomogenen Raten allein aus dem aktuellen Geschehen der Produktionslinie stromabwärts bestimmen lassen. Sofern bei der Berechnung der gesuchten Ergebnisse des Intervalls die Maschine der Arbeitsstation 7 als erstes und dann die weiteren Maschinen in Richtung stromaufwärts iterativ verarbeitet werden, lässt sich für jede Maschine M_i die $tbf_{2_i}(t)$ und die $ttr_{2_i}(t)$ ohne Wissen über die Vergangenheit bestimmen.

3.2.2 Unsicherheit innerhalb der Symbolsequenzen

Es gilt jedoch eine weitere Schwierigkeit zu lösen, denn das Modell kennt nur die Arbeitsstation, in der dem beobachteten Symbol nach gerade eine Maschine ausgefallen oder repariert worden ist, aber eben nicht die exakte Maschine. Sonst wäre es nicht nur

partiell beobachtbar. Dadurch kommt es zu Unsicherheiten bezüglich der Einträge der Outputmatrizen.

Dazu eine kurze Einführung in die Symbole von M. Für den Initialzustand aller Maschinen wird angenommen, dass sie funktionstüchtig sind. Für jede Maschine wird außerdem eine Sequenz an Symbolen O über alle Intervalle vorausgesetzt, die zu Anfang aussagt, dass alle Maschinen laufen. Die Zustände S_1 und S_2 senden unter Garantie das Symbol Maschine Funktionstüchtig, S_3 und S_4 signalisieren einen Ausfall der Maschine. Kommt es an einer Arbeitsstation zu einem Defekt, wechselt ein Symbol in einer Sequenz der Arbeitsstation, jedoch ist bei mehreren Maschinen in der Arbeitsstation nicht bekannt, welche Symbolsequenz genau betroffen ist. Die HMM lässt jedoch nur ein Symbol pro Intervall und Modell zu. Daher müssen die Werte der Outputmatrix B dynamisch angepasst werden, um den Entwicklungen der Unbekannten in der Symbolsequenzen gerecht zu werden.

Ein Beispiel hierfür ist die Arbeitsstation fünf, die mit vier Maschinen über größte Anzahl an Produktionsanlagen verfügt. Fällt eine davon aus, besteht für die letzte Arbeitsmaschine unabhängig von der Ausfallrate eine Chance von 1 zu 4, dass sie die betroffene ist. Unter der Voraussetzung, dass die Arbeitsstation stromabwärts korrekt arbeiten, beträgt demnach der Wert von B für S_1 0.75 und für S_3 0.25. Für die zweitletzte Maschine sehen die Werte jedoch wiederum anders aus. Hier beträgt, die Chance, dass das Symbol die Maschine stromabwärts betrifft 0.25, was dann der Wert von B für S_2 ist. S_1 hingegen ist nur noch zu 50% sicher und Möglichkeit, dass gerade betrachtete Maschine direkt betroffen ist und somit S_4 gilt, liegt wie schon zuvor bei $\frac{1}{4}$. Diese Veränderungen setzten sich auf die beiden letzten Maschinen fort.

Der Wert, der aus der Outputmatrix B für einen Zustand folgt, ist daher von der Anzahl an ausgefallenen Maschinen der Arbeitsstation abhängig, die die gerade betrachtete Maschine enthält, statt von einer statischen Wahrscheinlichkeit. Daher handelt es sich bei B im Rahmen dieser Arbeit um einen Vektor mit der Anzahl an Modellzuständen als Länge. Die Werte dieses Vektors sind zeitinhomogen und lassen sich über eine Baumstruktur wie bei einem mehrstufigen Zufallsexperiment bestimmen, bei dem aus einem Behälter Symbole entnommen werden, ohne sie zurückzulegen. Die Anzahl der Symbole entspricht der Anzahl der Maschinen in der Arbeitsstation. Wenn zudem eine Arbeitsstation stromabwärts existiert, geht deren Zustand, ob funktionstüchtig oder nicht, zusätzlich in den Behälter ein und wird unter Garantie zuerst gezogen. Die Wurzel der Baumstruktur

repräsentiert daher den Zustand der Maschinen flussabwärts. Um festzustellen, für welchen Zustand ein Blatt einen Wahrscheinlichkeitswert enthält, muss der bisher im Baum eingeschlagenen Pfad von der Wurzel bis zum Blatt ausgewertet werden. Enthalten mehrere Blätter für einen Zustand Wahrscheinlichkeitswert größer Null, so werden diese addiert. Die Summe der Einträge von B betragen für jedes Modell der Produktionslinie Eins. In der folgenden Grafik wird das eben geschilderte Vorgehen knapp visualisiert, wobei auf das Zeichnen von Blättern mit Werten gleich Null verzichtete wurde.

Symbole: $1 \cdot A + 1 \cdot I$
 A...funktionstüchtig
 I...funktionsuntüchtig

Arbeitsstation stromabwärts:

Letzte Maschine
 der betrachteten
 Arbeitsstation:

Vorletzte Maschine
 der betrachteten
 Arbeitsstation:

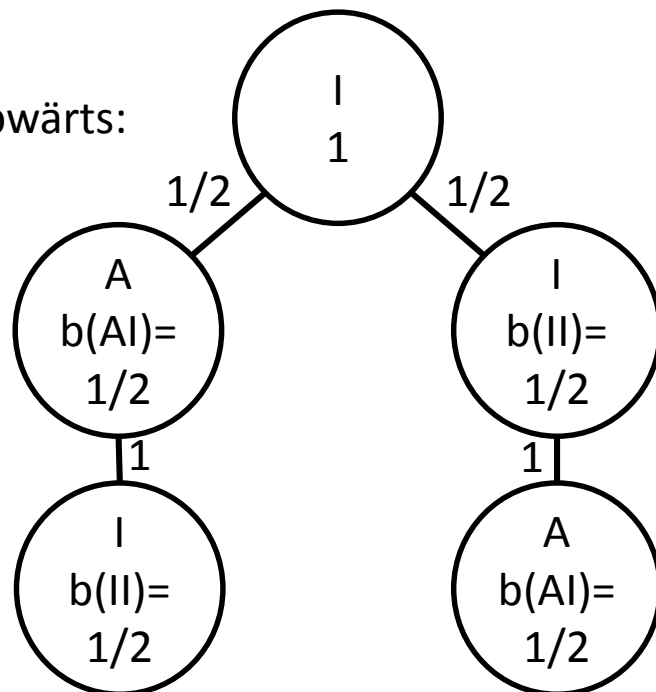


Abbildung 3.2 Beispiel zur Berechnung der Einträge der Outputmatrix

Zusammenfassend besteht daher das Modell M der Pizzaproduktionslinie aus einer Menge an Modellen $M = \{M_1, M_2, \dots, M_{14}\}$, wobei die Übergangsmatrix A_i für Modell M_1 bis M_{12} dynamisch ist und die Werte der Outputmatrix B_i sich bei jeder Änderung

3.2.3 Implementation des HMM

Bis auf die bereits beschriebene dynamische Werte der Einträge von B und und der tbf_{2i} und ttr_i , sowie der iterativen Verarbeitung der Modelle M_i aus der sortierten Menge, angefangen beim letzten Element, unterscheiden sich die Modelle selbst nicht bezüglich der Lösung der Anwendungsfälle Evaluation, Decodierung und Training.

Jedoch stellt der große Betrachtungszeitraum über vier Jahre und das kurze Intervall von nur einer Minute zwischen zwei Beobachtungen die Decodierung und das Training der Modelle der Pizzaproduktionslinie vor Herausforderungen.

Zum einen werden die berechneten Wahrscheinlichkeiten mit größer werdender Länge der Symbolsequenz sehr klein. Um die Werte weiterhin durch 64bit-Gleitkommazahlen repräsentierbar zu halten, werden diese logarithmisch skaliert. (vgl. Fink 2008, S. 120) Dadurch wird jedoch insbesondere das Addieren von Wahrscheinlichkeiten teuer für die Performance der Modelle, es existiert jedoch eine Methode, die diese Aufgabe löst, ohne die betroffene Wert potenzieren zu müssen. (vgl. Melnikoff 2003, S. 939–940)

Aus der Länge der Symbolsequenz folgen jedoch auch Probleme bezüglich des Speicherbedarfs. Im Falle des Trainings werden bei dem herkömmlichen, iterativen Vorgehen für jede der 1.271.520 beobachteten Symbole der Sequenz O ein kompletter Satz an Forward- und Backward-Wahrscheinlichkeiten auf einmal im Speicher gefordert. In dem vorliegenden Modell sind es nur 28 Zustände pro Symbol. Insgesamt wird also Speicherplatz für mindestens 71.205.120 reelle Werte benötigt.

Dieser Herausforderung lässt sich jedoch mit einem rekursiven Algorithmus für das Training lösen. Dieser hat bei einer Symbolsequenz der Länge n eine Zeitkomplexität von $O(n \log n)$ und einer Speicherkomplexität von $O(\log n)$. Damit ist die Zeitkomplexität der iterativen Variante mit $O(n)$ zwar unterlegen, doch der Speicherbedarf von $O(n)$ wird drastisch reduziert. (vgl. Buchholz 2012, S. 86–87)

Eine Reduzierung von solchen Umfang ist jedoch für dieses Modell nicht notwendig und der Zeitaufwand für Decodierung und Training wäre aufgrund dieses Verfahrens wenigstens um ein Faktor zehn höher als beim iterativen Verfahren. Deshalb wurden die Ansätze des rekursiven Verfahren und des iterativen Verfahren miteinander gemischt, um ein Kompromiss zwischen der Performance und der Reduktion des Speicherverbrauchs zu erreichen.

Dieser besteht darin, dass eine Intervalllänge k festgelegt wird, die für eine möglichst große Anzahl an Zeitschritten, für die ein paralleles Ablegen aller Forward- und Backward-Wahrscheinlichkeiten aller Zustände dieser Zeitschritte möglich ist.

Daraufhin wird der Forward-Algorithmus des Trainings einmal komplett durchgeführt, es werden aber nur die Werte für die Zeitschritte im RAM abgelegt, die sich ohne Rest durch k teilen lassen, sodass die Modellzustände aller n/k -Grenzen der Zeitintervalle bekannt

sind. Da Markovprozesse gedächtnislos sind, reichen diese aus, um von jeder unteren Intervallgrenze aus den Forward-Algorithmus für den vorliegenden Abschnitt der Symbolsequenz vom neuen starten zu lassen. Das Training des Modells wird dann für das letzte Intervall zuerst vorgenommen. Für dieses wird der Forward- und Backward-Algorithmus durchgeführt und die berechneten Werte werden wie in Abschnitt 2.2.5 verarbeitet. Dann wird ein Intervall nach vorn gesprungen und der Vorgang wiederholt sich, bis das erste Intervall ausgewertet worden ist.

Dieser modifizierte iterative Ansatz ist von der Zeitkomplexität her noch immer $O(n)$, es bedarf aber der zweimaligen Anwendung des Forward-Algorithmus. Dies ist jedoch noch immer bedeutend schneller als der rekursive Algorithmus besonders für sehr lange Symbolsequenzen. Der Speicherbedarf ist jedoch mit $O(k+n/k)$ auch dementsprechend höher.

Leider können die Modell übergreifende Strukturen aus Abschnitt 3.2.1 nur im ersten Trainingslauf einer HMM verwendet werden, da die Kanten des HMM aus Produkten bestehen, deren Werte unabhängig von ihren Faktoren optimiert werden. Somit werden die Beziehungen von Produkten zu deren Faktoren nach einem Trainingslauf aufgelöst. Die direkten Werte für die tbf und ttr einer Maschine lassen sich daher nur erahnen.

3.3 Aufbau und Implementation der CHnMM

ASPN lassen sich nicht so wie eine DTMC bei einem HMM einfach direkt übernehmen, da diese nicht Zustandsbasiert sind. Der Zustandsraum muss daher erst entwickelt werden. Da der aber genau wie bei der HMM zu umfangreich ausfällt, wenn die ganze Produktionslinie in einem Modell abgebildet wird, wird die Produktionslinie wieder entsprechend ihrer Maschinen modularisiert. Das Vorgehen ist dabei vergleichbar mit Abschnitt 3.2.1.

3.3.1 Struktur des Modells

Wie in Abschnitt 3.2.1 besteht das Modell der Produktionslinie aus der sortierten Menge an Modellen $M = \{M_1, M_2, \dots, M_{14}\}$, von denen jedes bis auf das letzte die Entwicklungen stromabwärts berücksichtigt. Deren dynamisches Verhalten soll anhand der Symbolsequenz $O = \{O_1, O_2, \dots, O_U\}$ rekonstruiert werden. Der Zustandsraum für M_{14} entspricht daher der Abbildung 3.4, während die restlichen Maschinen mit dem Zustandsraum $S = \{S_1, S_2, S_3, S_4\}$ dem Schema der Abbildung 3.3 folgen. Wie beider HMM steht dabei $S_1 = AA$ für die Markierung, dass sowohl die betrachtete Maschine i wie auch die Maschinen

stromabwärts korrekt arbeiten. Im Falle von $S_2 = AI$ ist mindestens eine der Maschinen stromabwärts defekt, während die betrachte Maschine zwar heile, aber inaktiv ist und daher nicht ausfallen kann. In der Markierung $S_3 = II$ ist die betrachtete Maschine und mindestens eine Maschine stromabwärts funktionsuntüchtig. $S_4 = IA$ repräsentiert den Umstand, das lediglich die betrachtete Maschine nicht korrekt arbeitet, im Gegensatz zum funktionstüchtigen Teil der Produktionslinie stromabwärts.

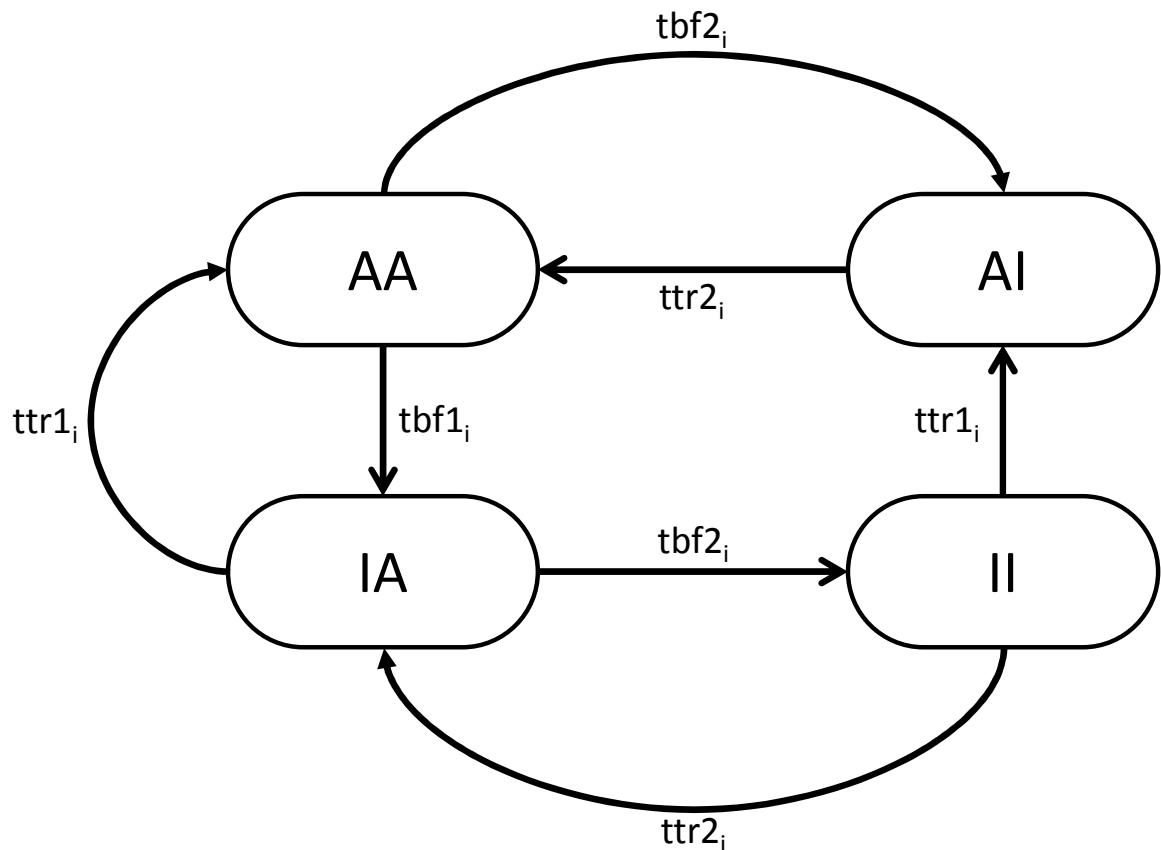


Abbildung 3.3 Zustandsraum der CHnMM einer Maschine

Ebenfalls der HMM entsprechend gilt, dass die Charakteristika der Aktivität von $tbf1_i$ aus der Tabelle 3.1 und von $ttr1_i$ aus der Tabelle 3.2 abzulesen sind, wobei es sich bei den Aktivitäten um die entsprechenden Weibullverteilungen handelt. Somit sind auch die Übergangsraten zwischen allen Markierungen nicht mehr zeithomogen.

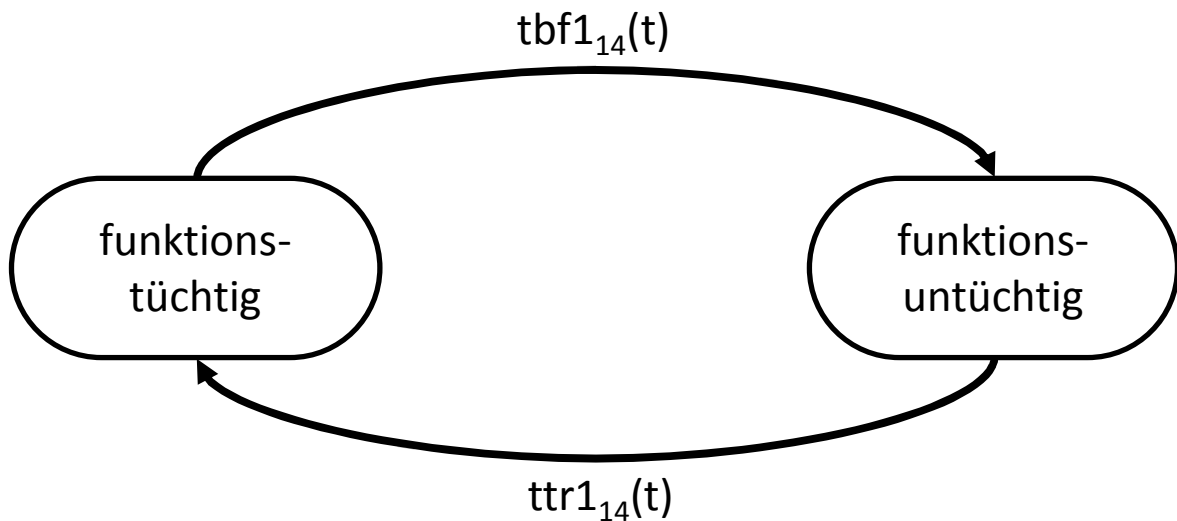


Abbildung 3.4 Zustandsraum der CHnMM der Arbeitsstation 7

Der Ansatz aus Abschnitt 3.2.1 zur Berechnung der Übergangsraten wird für die Aktivitäten $tbf2_i$ und $ttr2_i$ übernommen. Daher werden sie aus dem Verhalten der Maschinen stromabwärts der betrachteten Maschine i zum Zeitpunkt t , an dem das Symbol O_u beobachtet wurde, gefolgert. Statt aber die Parameter für Form und Skalen zu schätzen, werden $P_{sojourn}$ und P_{change} (siehe Abschnitt 2.3.2) der Aktivitäten direkt berechnet. Dabei wird auf die Proxel p der Menge $R_{i+1, u-1}$ des Nachfolgermodells M_{i+1} zugegriffen.

Im Falle des Forward-Algorithmus bildet sich $P_{sojourn}$ aus der Summe aller Produkte der Forward-Wahrscheinlichkeiten der Proxel $R_{i+1, u-1}$ im Zustand S_1 mit ihren jeweiligen $P_{sojourn}$, geteilt durch die Summe der gerade verwendeten Forward-Wahrscheinlichkeit. Bei der Berechnung von P_{change} wird der gerade erwähnte Zähler zum Nenner, während im Zähler selbst die P_{change} hinzukommt. Dabei werden jedoch nur Aktivitäten berücksichtigt, die aus der Markierung S_1 hinausführen, wie bei der P_{change} der $tbf2_i$, oder in diesen hineinführen, wie bei der P_{change} der $ttr2_i$. Bei der $P_{sojourn}$ der $tbf2_i$ werden demnach auch nur Proxel der Markierung S_1 berücksichtigt, während bei der $P_{sojourn}$ der $ttr2_i$ alle anderen Markierungen von Bedeutung sind. Somit gilt:

$$tbf2_i \cdot P_{sojourn} = \frac{\sum_{p \in R_{u-1} \wedge p.S=S_1} p \cdot \alpha * P_{sojourn}(p, t)}{\sum_{p \in R_{u-1} \wedge p.S=S_1} p \cdot \alpha}$$

$$ttr2_i \cdot P_{sojourn} = \frac{\sum_{p \in R_{i+1, u-1} \wedge p.S \in \{S_2, S_3, S_4\}} p \cdot \alpha * P_{sojourn}(p, t)}{\sum_{p \in R_{i+1, u-1} \wedge p.S \in \{S_2, S_3, S_4\}} p \cdot \alpha}$$

Formel 3.5 Psojourn der Maschinen stromabwärts

$$\begin{aligned}
 & \text{tbf}2_i \cdot P_{\text{change}} \\
 &= \frac{\sum_{p \in R_{u-1} \wedge p.S=S_1} p \cdot \alpha * P_{\text{sojourn}}(p, t) * (P_{\text{change}}(\text{tbf}1_{i-1}, t) + (P_{\text{change}}(\text{tbf}2_{i-1}, t)))}{\sum_{p \in R_{u-1} \wedge p.S=S_1} p \cdot \alpha * P_{\text{sojourn}}(p, t)} \\
 & \text{ttr}2_i \cdot P_{\text{change}} = \frac{\sum_{p \in R_{u-1} \wedge p.S=S_2} p \cdot \alpha * P_{\text{sojourn}}(p, t) * P_{\text{change}}(\text{ttr}2_{i-1}, t)}{\sum_{p \in R_{u-1} \wedge p.S=\{S_2, S_3, S_4\}} p \cdot \alpha * P_{\text{sojourn}}(p, t)} \\
 & \quad + \frac{\sum_{p \in R_{u-1} \wedge p.S=S_4} p \cdot \alpha * P_{\text{sojourn}}(p, t) * P_{\text{change}}(\text{ttr}1_{i-1}, t)}{\sum_{p \in R_{u-1} \wedge p.S=\{S_2, S_3, S_4\}} p \cdot \alpha * P_{\text{sojourn}}(p, t)}
 \end{aligned}$$

Formel 3.6 Pchange der Maschinen stromabwärts

Die Werte von P_{sojourn} und P_{change} der Aktivitäten $\text{tbf}2_i$ und $\text{ttr}2_i$ sind für alle Proxel gleich, da diese das gesamte Verhalten der Maschinen stromabwärts abbilden und es somit unerheblich ist, wie lange diese Aktivität im Proxel schon freigegeben ist. Die Berechnung dieser beiden Größen geht beim Backward-Algorithmus analog zum Forward-Algorithmus vonstatten, mit dem einzigen Unterschied, dass Backward-Wahrscheinlichkeiten die Forward-Wahrscheinlichkeiten ersetzen.

3.3.2 Unsicherheit innerhalb der Symbolsequenzen

Aus den gleichen Grund, wie er in Abschnitt 3.2.2 beschrieben wurde, sind die Werte der Outputmatrix dynamisch. Wie bei der HMM der Produktionslinie wird auch in diesem Fall ein Vektor für die Outputmatrix angenommen. Im Gegensatz zur HMM werden die Symbole jedoch durch abgeschlossene Aktivitäten ausgesendet, statt durch das verharren in einem bestimmten Zustand. Das Verfahren, wie es bei der HMM zur Bestimmung von $B_{i, u}$ zum Zeitpunkt des Symbols O_u verwendet wurde, ist daher nicht mehr direkt anwendbar. Von dessen Ergebnissen lässt sich jedoch auf die Outputmatrix $b_{i, u}$ des Modells M_i schließen. Da hier die Wahrscheinlichkeit der Aktivitätsabschlüsse in Abhängigkeit der Symbolsequenz gesucht werden, wird der Übergang von der Outputmatrix $B_{i, v}$ der Maschine i zum Zeitpunkt v der Beobachtung O_{u-1} zur Outputmatrix $B_{i, w}$ zum Zeitpunkt w der Beobachtung O_u untersucht.

Die Vorgehensweise ist dabei wie folg. Sowohl $B_{i, v}$ wie auch $B_{i, w}$ werden als Zustandsraum eines Markovprozesses angesehen, der der Abbildung 3.5 entspricht. Von einem Schritt des Prozesses sind der vorherige Zustand und der Folgezustand des Markovmodells bekannt. Die Übergangsraten werden folglich gesucht. Dafür wird folgendes Gleichungssystem aufgestellt, welches es zu lösen gilt:

$$B_{i,w}(AA) = B_{i,v}(AA)(1 - AAAI - AAIA) + B_{i,v}(AI)AIAA + B_{i,v}(IA)IAAA$$

$$B_{i,w}(AI) = B_{i,v}(AI)(1 - AIAA) + B_{i,v}(AA)AAAI + B_{i,v}(II)IIAI$$

$$B_{i,w}(II) = B_{i,v}(II)(1 - IIAI - IIIA) + B_{i,v}(IA)IAII$$

$$B_{i,w}(IA) = B_{i,v}(IA)(1 - IAII - IAAA) + B_{i,v}(II)IIIA + B_{i,v}(AA)AAIA$$

Formel 3.7 Gleichungssystem zum Lösen der Outputmatrix-Übergänge

Die Faktoren mit den vier Buchstaben ergeben die Wert von $b_{i,u}$ bezüglich des Übergangs von einer Markierung, die mit den ersten beiden Buchstaben des Faktors festgelegt wird, zu einer anderer, die durch die letzten beiden Buchstaben benannt wird. Die CHnMM der Pizza-Produktionslinie bricht dementsprechend mit der Vorgabe, dass ein Wert der Outputmatrix, einer Aktivität zugeordnet wird. Stattdessen enthalten diese Werte für die Kanten des Graphen des Zustandsraums.

Das Gleichungssystem aus Formel 3.7 ist jedoch nur mit Hilfe zusätzlicher Informationen möglich. Zum einen muss die Bedeutung des Symbols O_u berücksichtigt werden. Wenn dieses ausdrückt, dass bei einer Maschine gerade ein kritischer Fehler aufgetreten ist, sind die Faktoren AIAA, IAAA, IIIA und IIAI gleich Null. Das Gleichungssystem kann dann ohne weiteres gelöst werden.

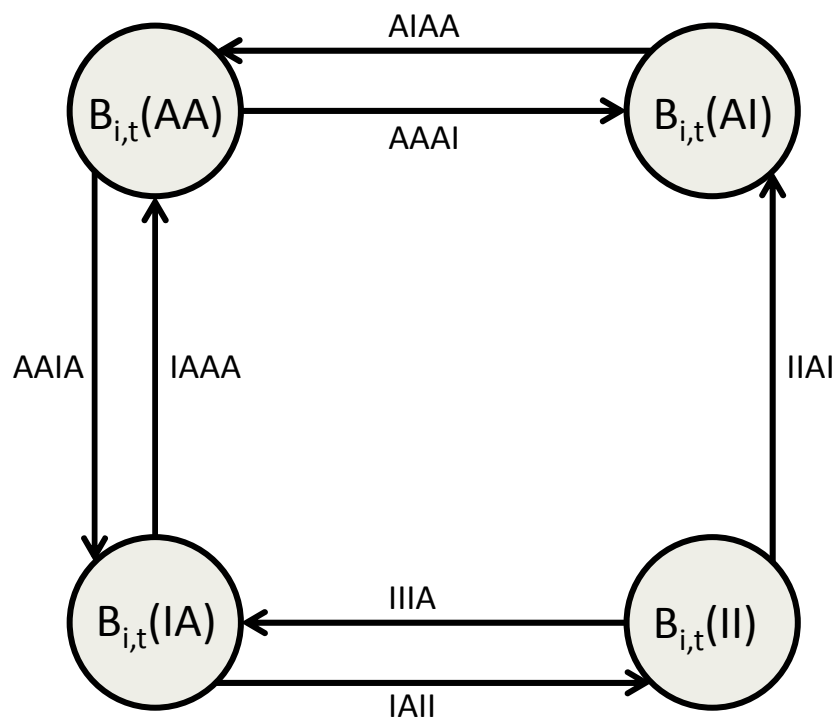


Abbildung 3.5 Markovprozess der Outputmatrizen

Wird jedoch signalisiert, dass eine Maschine nach einer Unterbrechung durch einen Defekt just wieder an der Produktion teilnimmt, werden zwar ebenfalls einige Faktoren auf Null gesetzt, nämlich AAI, AAIA und IAI, doch reicht diese Information nicht aus, da es noch immer unendliche viele Lösungen hat. Daher muss ein Faktor angenommen werden. Der Faktor AIAA wurde dabei als geeignet identifiziert, da dieser lediglich die Zustandsänderung von Werten bezüglich der Maschinen stromabwärts abbildet. Daher sollte er von der betrachteten Maschine i unabhängig zu bestimmen sein. Zu diesem Zweck wird das Markovmodell der Outputmatrix des Modells M_{i+1} wie in Abschnitt 3.2.1 ein Modell mit dem Zustandsraum $W_i = \{W_{1,i}, W_{2,i}\}$ mit $W_{1,i} = B_{i+1}(AA)$ und $W_{2,i} = B_{i+1}(AI) + B_{i+1}(II) + B_{i+1}(IA)$. AIAA ergibt sich dann durch Umstellen aus folgender Gleichung:

$$W_{1,i} = W_{1,i} + AAIA_i * W_{2,i}$$

Formel 3.8 Faktor AIAA

3.3.3 Implementation der CHnMM

Die Umsetzung der CHnMM für die Pizza-Produktionslinie erfordert neben dem Bruch mit der Vorgabe, dass eine Aktivität bezüglich eines Symbols stets den gleichen Wert aus der Outputmatrix erhält, noch eine weitere Missachtung einer Restriktion.

Denn eine CHnMM geht davon aus, dass, wenn ein Symbol empfangen wurde, auch eine Veränderung im abgebildeten System stattgefunden hat. Da es im Fall einer Arbeitsstation mit mehreren Maschinen jedoch unklar ist, welche Maschine das Symbol gesendet hat, kann diese Restriktion nicht mehr gehalten werden. Daher werden die Algorithmen zur Evaluation, Decodierung und zum Training dahin gehend geändert, dass ein Zustand während eines ausgegeben Signals auch im selben Zustand verbleiben kann. P_{change} wird in diesem Fall auf eins gesetzt.

Zudem wurde ursprünglich davon ausgegangen, dass die Aktivität $tbfl_i$ in der Markierung AI nicht altert. Diese Annahme sorgte bei den Experimenten jedoch für einen exponentiellen Anstieg der Menge der Proxel (vgl. Buchholz 2012, S. 98) und wurde daher verworfen.

Bezüglich der Speicherkomplexität steht ein CHnMM ähnlichen Herausforderungen gegenüber wie ein HMM. Zwar ist der Aufwand eines CHnMM nicht mehr vom Betrachtungsraum abhängig, sondern von der Länge Symbolsequenz, doch müssen nicht nur beim Training, sondern auch bei der Decodierung die auf die rekursive Variante der Algorithmen zur Lösung dieser Anwendungsfälle zurückgegriffen werden.

In dem vorliegenden Fall war dies bei der Decodierung jedoch nicht nötig, da mit direkten Links gearbeitet wurde, statt die Mengen an Proxel für jeden Eintrag der Beobachtungssequenz zu behalten. Java entsorgt daher automatisch die nicht gebrauchten Proxel und ermöglicht somit die iterative Decodierung selbst über die 3200 Symbole des Beobachtungszeitraums hinweg.

Beim Training wurde jedoch auf das modifizierte iterative Verfahren aus Abschnitt 3.2.3 übernommen. Dies ist trotz der fehlenden Gedächtnislosigkeit möglich, da Proxel alle relevanten Information zu Weiterführung ihres Weges enthalten. Sofern also ein entsprechender Satz an Proxel für einen Zeitpunkt vorhanden ist, kann von diesem aus auch der Forward-Algorithmus fortgeführt werden.

Die Weibullverteilung selbst stellt jedoch auch ein Problem dar, da kein verwendbarer Schätzer für den Form-Parameter im Zuge des Trainings einer CHnMM identifiziert werden konnte. Daher konnte lediglich der Skalen-Parameter angepasst werden. Zudem bedarf es bei einer logarithmischen Skalierung der Wahrscheinlichkeitswerte ein anderes Verfahren, als das die CHnMM momentan bietet, da in diesem eine Differenz ausgerechnet werden muss (vgl. Buchholz 2012, S. 96). Dies ist bei logarithmischen Skalierung nicht möglich.

3.4 Ergebnisanalyse

Rein von der Performance her ist die CHnMM interessant, vor allem wegen der Eigenschaften der Symbolsequenz. Sie geht über einen sehr langen Zeitraum, dokumentiert aber seltene Ereignisse. So manches Mal gibt es für Hunderte von Stunden nicht einen Eintrag. Wenn es dann aber mal zu einem Ausfall der Maschine kommt, ist dieser meist in weit weniger als einer Stunde behoben. Dies sind ungünstige Bedingungen für HMM, da diese regelmäßige Messungen voraussetzen, die über einen langen Zeitraum in einem kleinen Intervall betrachtete werden müssen. Die CHnMM jedoch reagiert explizit auf Ereignisse. Ihre Performance ist von der betrachteten Zeitspanne unabhängig und wird nur durch die Länge der Symbolsequenz bestimmt. Auch wenn die Aufteilung des Modells in kleinere Modelle viele teure Logarithmusadditionen zur Folge hatte, braucht der Forward-Algorithmus der CHnMM über die gesamte Zeitspanne von 1271520 Minuten weniger als die Hälfte der Zeit wie der Forward-Algorithmus der HMM.

Leider wurde festgestellt, dass der Forward so ziemlich die einzige Anwendung darstellt, bei der die in dieser Arbeit vorgestellte Aufteilung des Gesamtmodells in mehrere kleinere

anscheinend klappt. Die Produktionswahrscheinlichkeit entspricht der der ersten Maschine und kann daher einfach abgelesen werden. Wie sinnvoll jedoch die Ergebnisse sind, konnte nicht überprüft werden.

Das lag vor allem in dem Mangel an Modellen zum Vergleich. Eigentlich hätten diese durch das Training hergestellt werden können, doch dieses erbrachte weder beim HMM noch beim CHnMM sinnvolle Ergebnisse. Stattdessen senkte die lokale Optimierung der HMM die Werte der Kante von S_1 nach S_4 , anscheinend da es für das Modell hinsichtlich der Produktionswahrscheinlichkeit profitabler ist anzunehmen, der kritische Ausfall betreffe eine andere Maschine. Das Trainingsergebnis für die logarithmischen Einträge $a_{1,4}$ der Übergangsmatrizen A zeigt Tabelle 3.3.

HMM Training log0 Ergebnis	$a_{1,4}$ vor dem Training	$a_{1,4}$ nach dem Training
Arbeitsstation 1, Maschine 1	-9,63552	-43,660014585
Arbeitsstation 1, Maschine 2	-10,07522	-14,0219353288
Arbeitsstation 2, Maschine 1	-8,48218	-12,9061644247
Arbeitsstation 2, Maschine 2	-8,152805	-12,9367685038
Arbeitsstation 3, Maschine 1	-8,727548	-8,7752666501
Arbeitsstation 4, Maschine 1	-9,435124	-9,3915859131
Arbeitsstation 5, Maschine 1	-9,308991	-1102,1874594747
Arbeitsstation 5, Maschine 2	-9,605915	-104,2697188548
Arbeitsstation 5, Maschine 3	-9,408717	-86,4148055309
Arbeitsstation 5, Maschine 4	-9,736231	-91,8186832377
Arbeitsstation 6, Maschine 1	-9,119932	-737,8880305666
Arbeitsstation 6, Maschine 2	-9,457673	-68,6365150331
Arbeitsstation 6, Maschine 3	-9,883064	-84,7506628014

Tabelle 3.3 Trainingsergebnis HMM von S_0 nach S_4

Bei der CHnMM hingegen scheint es, die Verletzung der Restriktion, dass bei jedem Symbol auch eine Zustandsänderung sich geschehen muss, führte dazu, dass das Verweilen

im selbigen Zustand für die Modelle nicht nur attraktiv erscheint, sondern auch die geschätzten Erwartungswerte für $tbfl_i$ massiv steigert, da die Dauer der Aktivität direkt in die Schätzung deren Erwartungswerts hineinfließt. Dabei zeigt sich, dass sich das Modell dadurch nichts Gutes tut. Die Produktionswahrscheinlichkeit ist nach dem Training niedriger als zuvor. Einzig allein die Werte der ttr ergeben scheinbar Sinn, doch auch dort fällt auf, dass sich die Werte der Maschinen innerhalb einer Arbeitsstation auffällig ähneln. Dass die CHnMM grundsätzlich funktioniert, zeigen die relativ sinnvollen Ergebnisse in den Arbeitsstationen mit nur einer Maschine. Zu sehen ist das Ergebnis des Training des CHnMM in der Tabelle 3.4.

CHnMM Training Ergebnis	log0	rbfl _i vor Training	rtr1 _i vor Training	rbfl _i nach Training	rtr1 _i nach Training
Arbeitsstation 1, Maschine 1		11326,704	47.7922	563251,651	43,2098
Arbeitsstation 1, Maschine 2		23808,24	40.3000	563269,001	43,3104
Arbeitsstation 2, Maschine 1		4566	47.9492	420984,004	43,2161
Arbeitsstation 2, Maschine 2		2838	39.7905	420218,230	43,0687
Arbeitsstation 3, Maschine 1		10733,952	21.9036	7720,2713	22,33985
Arbeitsstation 4, Maschine 1		10789,008	31.2255	14414,1372	31,7544
Arbeitsstation 5, Maschine 1		13256,688	20.7965	688332,522	26,3064
Arbeitsstation 5, Maschine 2		9859,104	37.9651	688362,194	26,3064
Arbeitsstation 5, Maschine 3		14570,208	29.9020	688362,194	26,30644
Arbeitsstation 5, Maschine 4		8216,784	21.2500	688361,727	26,1331
Arbeitsstation 6, Maschine 1		11525,856	27.5362	658450,865	27,2509
Arbeitsstation 6, Maschine 2		14252,976	37.0000	658485,257	27,2509
Arbeitsstation 6, Maschine 3		13706,592	12.9365	658484,716	27,2509
Arbeitsstation7, Maschine 1		10733,952	42.9412	19219,1819	44,3749

Tabelle 3.4 CHnMM Training Ergebnisse

Auch die Decodierung funktioniert so nicht bei beiden Modellen, den es gelingt nicht, die Pfade der unterschiedlichen Modelle miteinander zu verknüpfen. So springt der Zustand eines Modells sowohl bei HMM wie auch bei der CHnMM nur zwischen den Zustand S_1

und S_2 hin und her, was semantisch gesehen über alle Maschinen der Produktionslinie hinweg gesehen unmöglich ist.

4 RÜCKBLICK UND AUSBLICK

4.1 Zusammenfassung

Diese Arbeit stellt eine Fallstudie zu Reliability Analysis eines technischen Systems mit Hilfe eines Hidden Conversive non-Markovian Model vor. Bei dem betrachteten System handelte es sich um eine Produktionslinie für Pizzen, einem Seriensystem, das über 14 Maschinen verfügt, die in sieben Arbeitsstationen aufgeteilt sind. Um aus das Modell nur partiell beobachtbar werden zu lassen, wurde die Annahme getroffen, dass ein Signal, gleich ob es den Defekt eine Maschine anzeigt oder deren Reparatur, nur der Arbeitsstation, aber nicht der Maschine zugeordnet werden kann. Jede dieser Maschinen wurde einzeln bewertet, wobei stets versucht wurde über Syntax die Semantik der Ergebnisse einer Maschine auch anderen mitzuteilen.

In diesem Zusammenhang wurde eine Strategie zu Aufteilung des Gesamtmodells entwickelt und umgesetzt. Zudem wurde der Aspekt des gemeinsamen Unbekannten mehrere Maschinen ebenfalls modelliert.

Jedoch kommt es nach Ausführung der Decodierung und des Trainings bei der HMM wie auch bei der CHnMM zu keinem sinnvollen Ergebnis. Die Sinnhaftigkeit der Ergebnisse des Forward-Algorithmus lassen sich nicht überprüfen, da Vergleichsmodelle fehlen, die das Training hätte liefern sollen, fehlen.

Doch die erzielten Ergebnisse offenbaren auch das Potenzial bezüglich des Konzepts, große Modelle in kleiner aufzuteilen. Der Zustandsraum beider Modelle umfasste lediglich in der Summe 28 Zustände. Dies sind sehr viel weniger, als die 2^{14} , die ein Modell über die gesamte Produktionslinie hätte. Daher ist das in dieser Arbeit vorgestellte Konzept einen zweiten blick wert.

Außerdem kommt sowohl für das Training der HMM, wie auch für das Training und die Decodierung der CHnMM ein Aufbau für die Lösungsalgorithmen entwickelt werden, der einen Kompromiss zwischen den schnellen, speicherlastigen, iterativen Verfahren und des langsamen, speicherschonenden rekursiven Verfahren darstellt.

Zudem erwies sich die CHnMM in diesem Fall als schneller gegenüber der HMM, was vor allem auf die Eignung der Symbolsequenz des betrachteten Systems zurückzuführen ist.

4.2 Ausblick

Daher bietet es sich bei weiteren Arbeiten zu diesem Thema an, an dieser Arbeit anzuknüpfen, doch im weiteren Versuch das Gesamtmodell der CHnMM so zu unterteilen, dass die Symbolmengen der Teilmodelle untereinander disjunkt sind. So kann sichergestellt werden, dass für jedes Modell bei jedem relevanten Symbol auch eine Veränderung des Systemzustands stattfindet und somit sämtliche Restriktionen der CHnMM eingehalten werden.

Zudem wurde das Verfahren zur Aufteilung des Gesamtmodells nur an einem konkreten Beispiel durchgeführt und nicht verallgemeinert. Auch hier besteht weiterer Forschungsbedarf.

LITERATURVERZEICHNIS

- Arnold, Dieter. *Handbuch der Logistik*. Berlin Heidelberg: Springer Verlag, 2008.
- Buchholz, Robert. *Conversive Hidden non-Markovian Models*. Magdeburg, 2012.
- Fink, Gernot A. *Markov Models for Pattern Recognition*. Berlin Heidelberg: Springer-Verlag, 2008.
- Gottschalk, G. „Systemtheorie in der Analytik.“ *Fresenius' Zeitschrift für analytische Chemie*, 1971: 2257-270.
- Koether, Reinhard. *Technische Logistik*. München: Carl Hanser Verlag, 2007.
- Krull, Claudia. *Discrete-Time Markov Chains: Advanced Applications in Simulation*. San Diego Erlangen: SCS Publishing House e.V., 2008.
- Kulkarni, V. G. *Introduction to Modeling and Analysis of Stochastic Systems*. New York Dordrecht Heidelberg London: Springer-Verlag, 2011.
- Lazzaroni, Massimo. *Reliability Engineering*. Berlin Heidelberg: Springer-Verlag, 2011.
- Liberopoulos, George. „Reliability Analysis of an Automated Pizza Production Line.“ *Journal of Food Engineering*, 2003: 1-25.
- Rabiner, Lawrence R. „A Tutorial on Hidden markov Models and Selected Applications in Speech Recognition.“ *Proceedings of the IEEE Vol. 70*. 1989. 257-286.
- Ryll, Frank. „Grundlagen der Instandhaltung.“ In *Instandhaltung technischer Systeme*, von Michael Schenk, 23-99. Berlin Heidelberg: Springer-Verlag, 2010.
- Trömel, Martin. „Das Wachstum technischer Systeme.“ *Natur & Wissenschaften*, 1995: 160-169.
- Zio, Enrico. *The Monte Carlo Simulation Method for System Reliability and Risk Analysis*. Berlin Heidelberg: Springer-Verlag, 2013.

Selbstständigkeitserklärung

Eidesstattliche Erklärung

Ich erkläre hiermit an Eides Statt, dass ich die vorliegende Arbeit selbstständig und ohne unerlaubte fremde Hilfe angefertigt, sowie andere als die angegebenen Quellen und Hilfsmittel nicht benutzt habe. Die aus fremden Quellen direkt oder indirekt übernommenen Stellen sind als solche kenntlich gemacht. Diese Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Magdeburg, 19.08.2013 (Mathias Vorbröcker)