Erweiterungen von Hidden-Markov-Modellen zur Analyse ökonomischer Zeitreihen

Inaugural-Dissertation
zur
Erlangung des Doktorgrades
der Mathematisch-Naturwissenschaftlichen Fakultät
der Universität zu Köln

vorgelegt von
Bernhard Knab
aus Worms

Köln 2000

Berichterstatter: Prof. Dr. Rainer Schrader
Prof. Dr. Ewald Speckenmeyer

Tag der mündlichen Prüfung: 13. Juli 2000

Kurzzusammenfassung

Hidden-Markov-Modelle (HMM) stellen eine allgemeine Methode in der statistischen Modellierung sequenzieller Daten oder Zeitreihen dar, die aufgrund ihrer großen Flexibilität und guten praktischen Handhabbarkeit in zahlreichen Anwendungsgebieten erfolgreich eingesetzt werden. In der vorliegenden Arbeit wird untersucht, inwieweit sich Hidden-Markov-Modelle zur Analyse und zur Simulation ökonomischer Zeitreihen eignen, die aus einzelnen Verträgen eines Bausparkollektivs stammen. Motiviert durch die Fragestellungen aus der Praxis präsentieren wir drei neue theoretische Erweiterungen der Hidden-Markov-Modellierung: ein HMM-basiertes Clusterverfahren, das eine Zuordnung der Trainingsdaten zu verschiedenen Modellen vornimmt und gleichzeitig die Modellparameter optimiert, Hidden-Markov-Modelle, deren Ausgaben gestutzten Normalverteilungen unterliegen, und eine erweiterte HMM-Klasse, mit der Abhängigkeiten innerhalb einer Zeitreihe modelliert werden können. Am Beispiel der Sparzahlungen in ein Bausparkollektiv können wir den praktischen Nutzen der Erweiterungen nachweisen.

Abstract

Hidden Markov models (HMMs) are a general method for statistical modelling of sequential data or time series. Due to their flexibility and efficiency they are successfully applied in many application areas. In this thesis we study the use of HMMs for the analysis and simulation of economical time series obtained from individual contracts of a building and loan association. We present three extensions of the HMM-theory arising from the requirements of our application: a cluster algorithm based on HMMs, solving the assignement problem for the training data and the models and optimizing the model parameters at the same time, the use of truncated normal densities as observation probability distributions and a class of extended HMMs for a modelling of dependencies within the time series. By modelling the saving payments of a building and loan association we can show the advantage of the extensions.

Inhaltsverzeichnis

1 Einleitung **1**

2 Hidden-Markov-Modelle (HMM) **5**

2.1 Definition eines HMM (diskrete Ausgaben) 6

2.2 Basisalgorithmen für Hidden-Markov-Modelle 8

2.2.1 Forward-Backward-Algorithmus 9

2.2.2 Viterbi-Algorithmus 11

2.2.3 Baum-Welch-Algorithmus 12

2.2.4 Konvergenz des Baum-Welch-Algorithmus 14

2.3 Hidden-Markov-Modelle mit stetigen Ausgaben 18

2.4 Skalierung 21

2.5 Reestimierung mit mehreren Sequenzen 23

2.6 Praktische Fragen, Modifikationen und Erweiterungen 25

3 Modellierung von Zeitreihen eines Bausparkollektivs **27**

3.1 Das Prinzip des Bausparens 27

3.2 Bisherige Modellansätze 29

3.2.1 Schichtenmodell 29

3.2.2 Mikrosimulationsmodell 30

3.2.3 Mesoskopisches Modell 30

3.3 Einsatz von Hidden-Markov-Modellen 33

3.3.1 Modellierungsidee 34

3.3.2 Auswahl der Daten und Sequenzen 34

3.3.3 Modellarchitektur 36

3.3.4 Training und Clusterung 37

3.3.5 Simulation 38

3.4 Erweiterungen für Hidden-Markov-Modelle 39

4 Clustern mit Hidden-Markov-Modellen 41
4.1 Problembeschreibung und Zielfunktion 41
4.2 Algorithmus 42
4.2.1 Maximum-Likelihood-Verfahren 43
4.2.2 Laufzeit 44
4.2.3 Wahl der Startmodelle 45
4.2.4 Modifikationen 46
4.3 Bewertung von Clusterungen 47
4.3.1 Zielfunktion 48
4.3.2 Abstandsmaße zwischen Hidden-Markov-Modellen 49
4.3.3 Datenverteilung auf den Zuständen bei stetigen Ausgaben 52
4.3.4 Modellierung und Simulationsgüte 55

5 HMM mit gestutzten Normalverteilungen 57
5.1 Allgemeine Herleitung der Reestimierungsformeln bei stetigen Ausgaben . . . 58
5.2 Einsatz von gestutzten Normalverteilungen 61
5.2.1 Notationen 61
5.2.2 Anpassung der Reestimierungsformeln 62
5.2.3 Numerische Berechnung der Reestimierungsparameter $\overline{\mu}$ und $\overline{\sigma}^2$ 66
5.2.4 Konsistenz mit Erwartungswert und Varianz 71
5.2.5 Praktische Probleme und Modifikationen 72
5.2.6 Erzeugung von Zufallszahlen 75
5.3 Alternative Dichtefunktionen 76

6 Erweitertes HMM mit Ausgabe-Klassen 79
6.1 Modellbeschreibung und Definitionen 80
6.2 Forward-Backward-Algorithmus 81
6.3 Baum-Welch-Algorithmus 84
6.4 Konvergenz des erweiterten Baum-Welch-Algorithmus 87
6.5 Viterbi-Algorithmus 89
6.6 Skalierung und mehrere Sequenzen 90

7 Anwendungen: Sparzahlungen eines Bausparkollektivs 91
7.1 Datensatz und relevante statistische Verteilungen 91
7.2 Architektur der verwendeten Modelle 94
7.3 Initialisierungen für das Clusterverfahren 97

7.4 Cluster-Indizes 99
7.4.1 Variation der Clusteranzahl 99
7.4.2 Variation der Anzahl der Zustände 101
7.5 Generierung von Sequenzen 102
7.5.1 Klassisches HMM (KL-HMM) 102
7.5.2 HMM mit Ausgabe-Klassen (AK-HMM) 109
7.5.3 Vergleich mit dem ***K***-means-Verfahren 118
7.6 Struktur einer Clusterung 120
7.7 Laufzeiten 122
7.8 Diskussion der Ergebnisse 123

8 Zusammenfassung und Ausblick **125**

A Reestimierungsformeln **127**

B Grenzwerte der Funktion $p(\mu)$ **131**

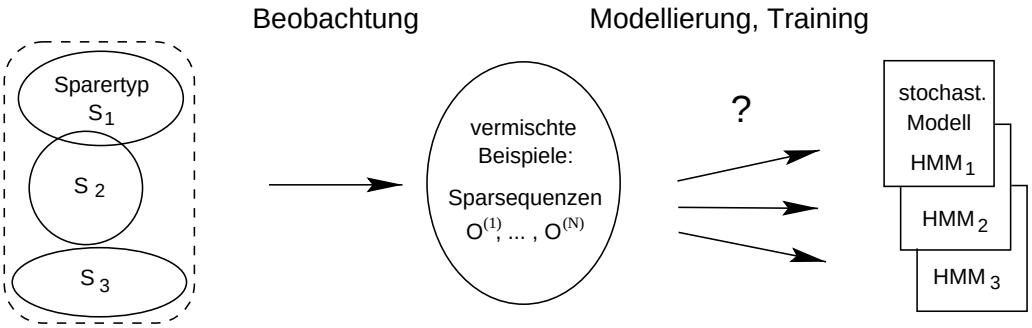
Literaturverzeichnis **133**

Danksagung **137**

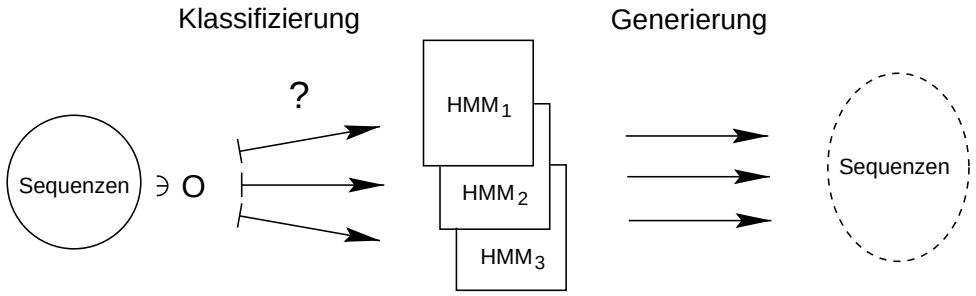
Kapitel 1

Einleitung

Klassifizierung und mathematische Modellierung bilden in vielen wissenschaftlichen Disziplinen die Grundlage zur Beschreibung natürlicher Vorgänge und Objekte. Einer Klassifizierung liegt die Vorstellung von der Existenz abstrakter Klassen zugrunde, die sich jedoch in den seltensten Fällen exakt definieren lassen. Statt dessen stehen dem Anwender Beispiele der Grundgesamtheit in Form von beobachteten Daten zur Verfügung.



Ein Ansatz, solche Daten mathematisch zu beschreiben, besteht darin, für jede abstrakte Klasse ein stochastisches Modell aufzustellen. Eine Klasse wird damit als ein den Daten zugrunde liegender stochastischer Prozess interpretiert und erfordert eine entsprechende Parameteranpassung, die auch als Training bezeichnet wird. In vielen Anwendungen kann dazu auf eine Datenmenge zurückgegriffen werden, deren Klasse bekannt ist, z. B. wenn für die Erkennung handgeschriebener Zeichen ein Modell eines bestimmten Buchstabens mit Schriftmustern verschiedener Personen trainiert wird. In anderen Fällen dagegen ist die Klassenzugehörigkeit der Trainingsdaten unklar, so dass bei der Modellbildung zusätzlich entschieden werden muss, welche Daten in welches Modell einfließen sollen. Mit Hilfe der trainierten stochastischen Modelle kann jedes weitere beobachtete Datum durch einen Vergleich der Wahrscheinlichkeiten, mit denen es von den jeweiligen Prozessen abstammt, klassifiziert werden. Schließlich können die Modelle zur Simulation bzw. zum Generieren künstlicher Daten eingesetzt werden.



Unter den stochastischen Modellierungsansätzen gewinnen Hidden-Markov-Modelle (HMM) immer mehr an Bedeutung, da sie aufgrund ihrer großen Flexibilität und ihrer guten praktischen Handhabbarkeit in zahlreichen Anwendungsgebieten äußerst erfolgreich eingesetzt werden.

In der vorliegenden Arbeit untersuchen wir, inwieweit sich Hidden-Markov-Modelle zur Analyse und zur Simulation ökonomischer Zeitreihen eignen, die aus einzelnen Verträgen eines Bausparkollektivs stammen. Hintergrund dieser Anwendung ist eine langjährige Kooperation zwischen den Landesbausparkassen und dem ZAIK, in deren Mittelpunkt die Entwicklung von Simulationsmodellen zur Kollektivanalyse und darauf aufbauender Prognosen steht. Diese stellen einen wichtigen Beitrag zur Liquiditäts- und Zuteilungsplanung sowie zur Produktpflege und -entwicklung dar.

Motiviert durch die praktischen Fragestellungen präsentieren wir drei neue theoretischen Erweiterungen der Hidden-Markov-Modellierung: ein HMM-basiertes Clusterverfahren, das das oben erwähnte Problem der Zuordnung der Trainingsdaten zu den Modellklassen löst, Hidden-Markov-Modelle, deren Ausgaben gestutzten Normalverteilungen unterliegen, und eine neue HMM-Klasse, mit der Abhängigkeiten innerhalb einer Zeitreihe flexibler modelliert werden können. Am Beispiel der Sparzahlungen in ein Bausparkollektiv können wir den praktischen Nutzen der Erweiterungen nachweisen.

Die Arbeit gliedert sich wie folgt:

Im zweiten Kapitel geben wir einen Überblick über die Theorie der Hidden-Markov-Modelle und stellen die im Weiteren relevanten Algorithmen und Methoden vor. Im dritten Kapitel gehen wir zunächst auf die Besonderheiten der Daten eines Bausparkollektivs und auf die bisher eingesetzten Simulationsmodelle ein. Für die Daten, die aus den einzelnen Verträgen einer Bausparkasse stammen, schlagen wir einen allgemeinen Modellierungsansatz auf der Basis von Hidden-Markov-Modellen vor.

Die folgenden drei Kapitel sind den HMM-Erweiterungen gewidmet. In Kapitel vier formulieren wir ein Clusterverfahren, bei dem eine Menge von Daten nach stochastischen Optimalitätskriterien in verschiedene, durch Hidden-Markov-Modelle abgebildete Gruppen eingeteilt wird, und diskutieren Bewertungsverfahren für eine solche Clusterung. Zur Abbildung der positiven Spargeld-Zeitreihen passen wir in Kapitel fünf den Trainingsalgorithmus für die Ausgabeparameter eines HMM an Dichtefunktionen von linksseitig gestutzten Normalverteilungen an; dabei entstehen analytisch nicht auflösbare Gleichungen, die jedoch in der Praxis numerisch gelöst werden können. Im sechsten Kapitel entwickeln wir schließlich eine neue Klasse von Hidden-

Markov-Modellen, die durch eine geeignete Modifikation der Übergangsparameter eines klassischen HMM entsteht. Die entsprechenden Basisalgorithmen können wir auf einfache Weise auf die erweiterten Modelle übertragen, ohne dass sich die Komplexität der Algorithmen dadurch ändert.

Im siebten Kapitel setzen wir die theoretisch erarbeiteten Verfahren und Modellerweiterungen für die Analyse von Spargeldeingängen einer Bausparkasse basierend auf Realdaten ein. Wir stellen verschiedene HMM-Strukturen, Initialisierungen und Modellklassen gegenüber und zeigen, dass die Ergebnisse durch die Modellerweiterungen deutlich verbesssert werden können.

Das achte Kapitel beschließt die Arbeit mit einer Zusammenfassung der Ergebnisse und einem Ausblick auf weitere Entwicklungsmöglichkeiten in Theorie und Anwendung.

Kapitel 2

Hidden-Markov-Modelle (HMM)

Hidden-Markov-Modelle stellen eine allgemeine Methode in der statistischen Modellierung sequenzieller Daten oder Zeitreihen dar. Bereits Mitte der sechziger Jahre entwickelt, werden sie seit den siebziger Jahren erfolgreich in der automatischen Spracherkennung eingesetzt. In den darauffolgenden Jahren fanden Hidden-Markov-Modelle Eingang in andere Anwendungsgebiete und erwiesen sich auch dort als äußerst wertvoll. Ihre große Stärke liegt darin, dass die Modelle einerseits für die Beschreibung einer Vielzahl von verschiedenartigen stationären und nichtstationären Prozessen geeignet sind. Anderseits stellen sie in der Praxis handhabbare und gut funktionierende Werkzeuge dar, für die effiziente Algorithmen und Berechnungsmethoden vorliegen.

Eine sehr gute und allgemein verständliche Einführung in die HMM-Theorie mit ausführlichen Literaturverweisen wird in [38] gegeben. Die darin ebenfalls zu findenden Anwendungsbeispiele sind wie die Mehrzahl der entsprechenden Veröffentlichungen in der Spracherkennung angesiedelt. Mit einem HMM-basierten Erkenner kann mit hoher Genauigkeit herausgefunden werden, welchem Wort bzw. welcher Wortfolge ein aufgezeichnetes und anschließend in Form einer Sequenz kodiertes Sprachsignal entspricht [4,17,28,40]. In der Bioinformatik werden Hidden-Markov-Modelle eingesetzt, um DNA- bzw. Proteinsequenzen zu analysieren. Dabei geht es meist darum, unterschiedliche funktionale Regionen zu erkennen (wie z. B. beim Suchen nach Genen) bzw. „ähnliche“ oder „verwandte“ Sequenzen zu finden, um so bereits bekannte Informationen ausnutzen und Vorhersagen über Struktur oder Funktion machen zu können. Einen guten Einstieg in diese Anwendungen bieten z. B. [9] und [10]. Beiden Disziplinen gemeinsam ist die Tatsache, dass sich in den Sequenzen jeweils individuelle Variationen bzw. Mutationen niederschlagen, die durch die Hidden-Markov-Modelle gut abgebildet werden können. Weitere Anwendungsgebiete sind beispielsweise die Handschriftenerkennung oder die Analyse diskreter Zeitreihen [31] (vgl. Abschnitt 3.3).

Wir stellen in den folgenden Abschnitten zunächst die formale Beschreibung eines HMM vor und erläutern dann die wichtigsten Methoden und Algorithmen der Hidden-Markov-Modellierung. Dabei gehen wir neben den theoretischen Aspekten auch auf praktische Implementierungsdetails ein. In Abschnitt 2.6 verweisen wir schließlich kurz auf einige der zahlreichen Variationen und Erweiterungsmöglichkeiten im Rahmen der HMM-Theorie.

2.1 Definition eines HMM (diskrete Ausgaben)

Ein Hidden-Markov-Modell (HMM) beschreibt einen stochastischen Prozess, der sich aus zwei gekoppelten Mechanismen zusammensetzt: Eine „versteckte“ Markov-Kette mit einer endlichen Anzahl von Zuständen wird in diskreten Zeitschritten durchlaufen und generiert dabei in jedem Zustand ein Ausgabesymbol gemäß einer von dem jeweiligen Zustand abhängenden Zufallsfunktion. Für einen Beobachter ist nur die so entstehende Sequenz von Ausgabesymbolen sichtbar, während die darunterliegende Folge von Zuständen verborgen bleibt.

Abbildung 2.1 zeigt ein Beispiel für ein einfaches HMM. Die Knoten des Graphen stehen für die möglichen Zustände des Markov-Prozesses und die gerichteten Kanten entsprechen den Übergängen zwischen zwei Zuständen, wobei jede Kante mit der jeweiligen Übergangswahrscheinlichkeit gewichtet ist. Die zusätzlichen Eingangskanten unter den Knoten enthalten die Wahrscheinlichkeit, dass der Prozess in dem entsprechenden Zustand startet. Zu jedem Zustand gehört eine diskrete Zufallsfunktion über einem gemeinsamen Ausgabealphabet, nach deren Verteilung die Symbole ausgegeben werden.

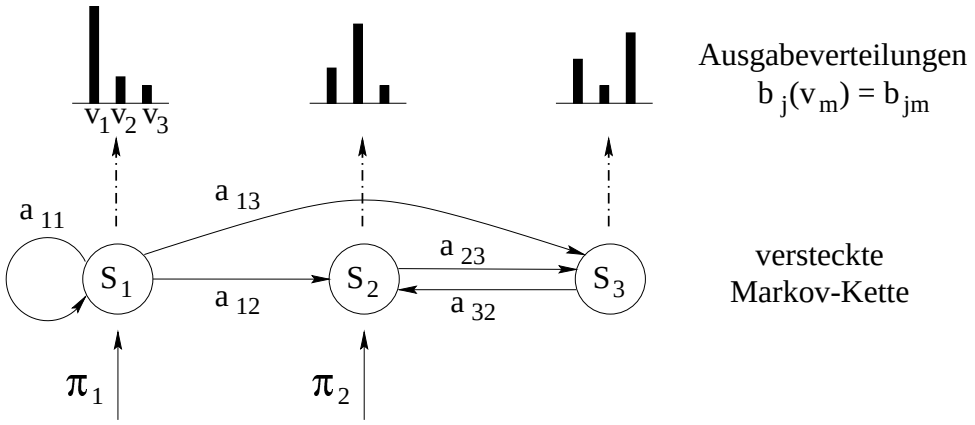


Abbildung 2.1: Beispiel für ein einfaches HMM mit diskreten Ausgaben

In der Regel werden bei einem HMM die folgenden Unabhängigkeitsannahmen getroffen:

1. Die Wahrscheinlichkeit, von einem Zustand zum nächsten zu wechseln (Übergangswahrscheinlichkeit), hängt nur von diesen beiden Zuständen und nicht von den Vorgängern ab.
2. Die Ausgabe bei gegebenem Zustand ist unabhängig von allen anderen Zuständen und Ausgaben.
3. Die Markov-Kette ist zeithomogen, d. h. die Übergangswahrscheinlichkeiten hängen nicht vom Zeitpunkt *t* des Zustandswechsels ab.

Die erste Annahme wird auch als Markov-Annahme oder Markov-Eigenschaft bezeichnet [27, 36], wobei in der HMM-Literatur meist von einer Markov-Kette erster Ordnung gesprochen

wird. Ein HMM, das den drei Unabhängigkeitsannahmen genügt, bezeichnen wir auch als klassisches HMM. Im Allgemeinen kann einem HMM eine komplexere Markov-Kette zugrunde liegen (siehe z. B. [9] und vgl. Abschnitt 2.6); wir betrachten im Folgenden jedoch nur klassische Hidden-Markov-Modelle.

Je nach Definition der Menge der Ausgabesymbole und deren erzeugenden Zufallsfunktionen auf den Zuständen sprechen wir von einem HMM mit diskreten oder mit stetigen Ausgaben. Wir wollen uns zunächst dem diskreten HMM zuwenden und an diesem die grundlegenden Prinzipien und Algorithmen erläutern, um später die Variante eines Modells mit stetigen Ausgaben vorzustellen.

Zur formalen Beschreibung eines HMM mit diskreten Ausgaben verwenden wir folgende Notationen:

N	Anzahl der Zustände in dem Modell
$\{S_1, \ldots, S_N\}$	Menge aller Zustände
$\pi = (\pi_1, \ldots, \pi_N)$	Vektor der Startwahrscheinlichkeiten des Prozesses für jeden Zustand
$A = \{a_{ij}\}$	$N \times N$-Matrix der Übergangswahrscheinlichkeiten von Zustand S_i nach Zustand S_j
M	Anzahl der möglichen Ausgabesymbole bzw. Größe des Ausgabealphabets
$\{v_1, \ldots, v_M\}$	diskretes Ausgabealphabet
$B = \{b_{jm}\}$	$N \times M$-Matrix der Ausgabewahrscheinlichkeiten mit $b_{jm} = b_j(v_m)$ als der Wahrscheinlichkeit für die Erzeugung des Ausgabesymbols v_m in Zustand j
T	Länge einer beobachteten Sequenz
$O = O_1 \cdots O_T$	beobachtete Sequenz von Ausgabesymbolen mit $O_t \in \{v_1, \ldots, v_M\}$
$Q = q_1 \cdots q_T$	interne Zustandsfolge bei der Ausgabe einer Sequenz O mit $q_t \in \{1, \ldots, N\}$

Für die Elemente der Matrizen A und B und des Verteilungsvektors π gilt

$$\begin{aligned} \pi_i &= P(q_1 = i) && 1 \leq i \leq N\,, \\ a_{ij} &= P(q_{t+1} = j | q_t = i) && 1 \leq i,j \leq N\,,\ t \in \{1, \ldots, T-1\}\,, \\ b_{jm} &= P(O_t = v_m | q_t = j) && 1 \leq j \leq N\,,\ 1 \leq m \leq M\,,\ t \in \{1, \ldots, T\}\,, \end{aligned}$$

wobei wir im Weiteren mit $P(E)$ die Wahrscheinlichkeit für ein Ereignis E bezeichnen [35]. Die Matritzen A und B sind demnach sogenannte stochastische Matrizen, und π stellt einen Verteilungsvektor dar; d. h. die Elemente von A, B und π sind nichtnegativ und es gilt

$$\sum_{j=1}^{N} \overline{\pi}_j = \sum_{j=1}^{N} \overline{a}_{ij} = \sum_{m=1}^{M} \overline{b}_{im} = 1\,, \quad i \in \{1, \ldots, N\}\,.$$

Diese Anforderungen bezeichnen wir auch als stochastische Nebenbedingungen.

Die Matrix A der Übergangswahrscheinlichkeiten entspricht der Adjazenzmatrix des Modellgraphen, wobei der Eintrag a_{ij} das Kantengewicht auf der Kante (i, j) darstellt (vgl. Abbildung 2.1). Eine Übergangs- oder Startwahrscheinlichkeit von null bedeutet, dass die entsprechende Kante

im Graphen nicht vorhanden ist. Somit bestimmen die positiven Einträge von A und π zusamen mit ihrer Größe N, die die Anzahl der Zustände festlegt, die Topologie oder die Struktur des Modells. In der Praxis wird einem HMM häufig ein ausgezeichneter Endzustand hinzugefügt, bei dem der stochastische Prozess stoppt (siehe Abschnitt 3.3.3).

Mit den Größen N, M, π, A und B ist ein HMM mit diskreten Ausgaben vollständig spezifiziert und für diese komplette Parametermenge hat sich die kompakte Schreibweise

$$\lambda := (\pi, A, B)$$

eingebürgert. Ein HMM kann also mit seiner Parametermenge identifiziert werden, und wir werden im Folgenden der Einfachheit halber je nach Kontext mit λ das Modell selbst oder dessen Parameter bezeichnen.

2.2 Basisalgorithmen für Hidden-Markov-Modelle

Der Einsatz von Hidden-Markov-Modellen zur stochastischen Modellierung in der Praxis erfordert effiziente Rechenmethoden und Algorithmen zur Anpassung der Modelle an die jeweiligen Probleme bzw. zu deren Verwendung als Simulationsmodelle. Im Wesentlichen tauchen in diesem Zusammenhang folgende drei Basis-Probleme auf [38]:

1. Gegeben eine Sequenz $O = O_1 O_2 \cdots O_T$ und ein Modell λ, wie groß ist die Wahrscheinlichkeit, dass das Modell λ eine Sequenz O erzeugt, und wie kann diese Wahrscheinlichkeit $P(O|\lambda)$ effizient berechnet werden?
2. Gegeben eine Sequenz O und ein Modell λ, welche Zustandsfolge $Q = q_1 \ldots q_T$ ist zur gegebenen Sequenz O in einem gewissen Sinne „optimal“?
3. Gegeben eine Sequenz O und ein Modell mit den Startparametern λ, wie können die Parameter des Modells so estimiert bzw. trainiert werden, dass $P(O|\lambda)$ maximal wird?

Bei allen drei Fragen wird implizit vorausgesetzt, dass die Architektur des verwendeten Modells, d. h. die Anzahl der Zustände und die möglichen Übergänge und Startzustände, schon festgelegt ist. In der Tat liegt aber gerade hier eine Schwierigkeit bei der Anwendung von Hidden-Markov-Modellen: es sollte schon eine grobe Vorstellung von der Modellarchitektur vorhanden sein, die zu dem jeweiligen praktischen Problem passt. Auf die damit verbundenen Fragen, Probleme und Lösungsansätze werden wir später noch ausführlich eingehen.

Die Wahrscheinlichkeit $P(O|\lambda)$ in Problem 1 soll vor allem die Frage beantworten, wie gut ein Modell zu einer Sequenz passt. Auf diese Größe werden wir zurückgreifen, wenn es darum geht, für eine Sequenz ein repräsentatives Modell unter mehreren möglichen auszuwählen. Das dritte Problem ist entscheidend bei der Verwendung von Hidden-Markov-Modellen zur Beschreibung von Datensequenzen aus der realen Welt, denn über das Trainieren der Modellparameter können die Modelle erst den zugrundegelegten Daten angepasst werden.

In den folgenden Abschnitten werden Algorithmen vorgestellt, die die angesprochenen Basis-Probleme lösen. Dabei wird sich zeigen, dass die Fragestellungen eng miteinander verknüpft sind. Die Darstellung folgt weitgehend [38].

2.2.1 Forward-Backward-Algorithmus

Unser Ziel ist die Berechnung der Wahrscheinlichkeit $P(O|\lambda)$. Diese kann als Summe über alle Zustandsfolgen Q der Sequenzlänge T dargestellt werden, und mit den in Abschnitt 2.2 getroffenen stochastischen Unabhängigkeitsannahmen gilt dann:

$$\begin{aligned} P(O|\lambda) &= \sum_{\text{alle } Q} P(O,Q|\lambda) \\ &= \sum_{\text{alle } Q} P(O|Q,\lambda)\, P(Q|\lambda) \\ &= \sum_{\text{alle } Q} \pi_{q_1} b_{q_1}(O_1) a_{q_1 q_2} b_{q_2}(O_2) \ldots b_{q_T}(O_T)\,. \end{aligned} \tag{2.1}$$

Die Anzahl der verschiedenen Zustandspfade beträgt im schlechtesten Fall N^T, wie Abbildung 2.2 verdeutlicht. Damit liegt aber die Komplexität der obigen Berechnung bei $O(TN^T)$.

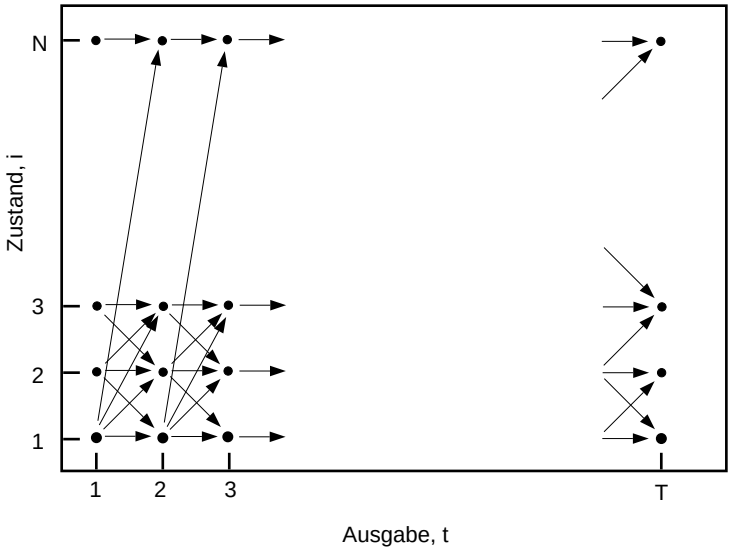


Abbildung 2.2: Pfade durch den Zustandsraum

Der sogenannte Forward-Algorithmus stellt eine effizientere Alternative zur Bestimmung von $P(O|\lambda)$ dar. Wir definieren zunächst die Forward-Variable $\alpha_t(i)$ als die Wahrscheinlichkeit, dass die Teilsequenz $O_1 O_2 \cdots O_t$ ausgegeben wird und sich die Zustandsfolge zum Zeitpunkt t in Zustand S_i befindet, gegeben die Modellparameter λ:

$$\alpha_t(i) := P(O_1 O_2 \cdots O_t, q_t = i|\lambda)\,. \tag{2.2}$$

Diese Variablen lassen sich induktiv berechnen:

1. Initialisierung, $t = 1$:

$$\alpha_1(i) = \pi_i b_i(O_1)\,, \qquad 1 \le i \le N\,. \tag{2.3a}$$

2. Iteration für $t = 1, \ldots, T-1$:

$$\alpha_{t+1}(i) = \left[\sum_{j=1}^{N} \alpha_t(j) a_{ji} \right] b_i(O_{t+1}), \quad 1 \leq i \leq N. \tag{2.3b}$$

Nachdem auch die Forward-Variablen für den letzten Zeitschritt T berechnet wurden, ergibt sich die gesuchte Wahrscheinlichkeit als die Summe über die $\alpha_T(i)$:

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

Abbildung 2.3 verdeutlicht die Induktion: Die Variable $\alpha_{t+1}(j)$ setzt sich aus der Summe der zuvor bestimmten $\alpha_t(i)$, multipliziert mit der jeweiligen Übergangswahrscheinlichkeit, zusammen. Pro berechnetem $\alpha_t(i)$ sind jeweils nur $2N + 1$ Operationen nötig, was zu einer Gesamtkomple-

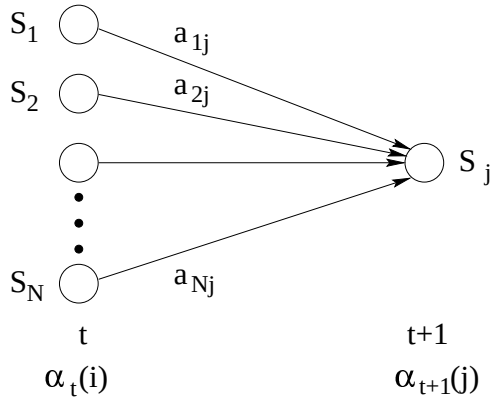


Abbildung 2.3: Rekursive Berechnung der $\alpha_{t+1}(i)$

xität von $O(TN^2)$ für den Forward-Algorithmus und somit für die Berechnung von $P(O|\lambda)$ führt. Mit Blick auf Abbildung 2.2 wird diese Verbesserung deutlich: Statt über alle Pfade des Zustandsraums zu summieren, werden in der Variablen $\alpha_t(i)$ die Wahrscheinlichkeiten aller Teilpfade vereint, die zum Zeitpunkt t in Zustand i enden.

Um das dritte Basisproblem effizient lösen zu können, definieren wir uns analog zu der Forward-Variablen $\alpha_t(i)$ jetzt die Backward-Variable $\beta_t(i)$. Diese bezeichnet die Wahrscheinlichkeit, dass die Teilsequenz $O_{t+1}O_{t+2} \cdots O_T$ ausgegeben wird, gegeben Zustand S_i zum Zeitpunkt t und das Modell λ:

$$\beta_t(i) := P(O_{t+1} \cdots O_T | q_t = i, \lambda). \tag{2.4}$$

Diese Variablen werden wir benutzen, um die Modellparameter mit einer vorgegebenen Sequenz zu trainieren, d. h. zum Lösen des Problems 3. Die $\beta_t(i)$ lassen sich ähnlich wie die Forward-Variablen iterativ mittels des Backward-Algorithmus berechnen:

1. Initialisierung, $t = T$:

$$\beta_T(i) = 1\,, \qquad 1 \leq i \leq N\,. \tag{2.5a}$$

2. Iteration für $t = T-1, T-2, \ldots, 1$:

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)\,, \quad 1 \leq i \leq N\,. \tag{2.5b}$$

Auch hier werden die Variablen entlang der Gitterstruktur des Zustandsraums (vgl. Abbildung 2.2) berechnet und die Komplexität des Algorithmus beträgt $O(TN^2)$.

Abschließend sei noch bemerkt, dass das Produkt $\alpha_t(i)\beta_t(i)$ der Wahrscheinlichkeit entspricht, die vollständige Sequenz O zu beobachten und zur Zeit t in Zustand i zu sein, und somit gilt für ein beliebiges $t \in \{1, \ldots T\}$:

$$\begin{aligned} P(O|\lambda) &= \sum_{i=1}^{N} P(q_t = i, O|\lambda) \\ &= \sum_{i=1}^{N} \alpha_t(i)\beta_t(i)\,. \end{aligned} \tag{2.6}$$

2.2.2 Viterbi-Algorithmus

Die zu Beginn des Abschnitts 2.2 gestellte Frage nach einer „optimalen“ Zustandsfolge Q bei gegebener Sequenz O (Problem 2) hängt natürlich von der Definition der Optimalitätsbedingung ab. Der Viterbi-Algorithmus dient dazu, diejenige Zustandsfolge zu ermitteln, die die Wahrscheinlichkeit $P(Q, O|\lambda)$ – und damit auch $P(Q|O, \lambda) = P(Q, O|\lambda)/P(O|\lambda)$ – maximiert. Die gefundene Zustandsfolge wird auch Viterbi-Pfad genannt.

Zu einer Sequenz $O = O_1 \cdots O_T$ und einer Zustandsfolge $Q = q_1 \cdots q_T$ definieren wir die Variable $\delta_t(i)$ folgendermaßen:

$$\delta_t(i) := \max_{q_1, q_2, \ldots, q_t} P(q_1 q_2 \cdots q_t = i, O_1 O_2 \cdots O_t|\lambda)\,, \tag{2.7}$$

d. h. $\delta_t(i)$ entspricht der maximalen Wahrscheinlichkeit, über einen Teilpfad $Q_{(t)} = q_1 \cdots q_t$ die Teilsequenz $O_{(t)} = O_1 \cdots O_t$ auszugeben und zur Zeit t im Zustand i zu sein, gegeben λ. Per Induktion folgt

$$\delta_{t+1}(j) = [\max_{1 \leq i \leq N} \delta_t(i) a_{ij}] \cdot b_j(O_{t+1})\,, \tag{2.8}$$

und somit können wir mit Hilfe der $\delta_t(i)$ iterativ die gesuchte maximale Wahrscheinlichkeit berechnen. Um jedoch am Ende den gesuchten Viterbi-Pfad rekonstruieren zu können, benötigen wir eine zusätzliche Variable $\psi_t(i)$, mit der wir uns einen Zustand merken, der (2.8) maximiert. Der komplette Viterbi-Algorithmus bestimmt den Viterbi-Pfad Q^* mit der zugehörigen Wahrscheinlichkeit P^* und verläuft wie folgt:

1. Initialisierung ($t = 1$):

$$\begin{aligned} \delta_1(i) &= \pi_i b_i(O_1)\,, \qquad 1 \leq i \leq N\,, \\ \psi_1(i) &= 0\,. \end{aligned}$$

2. Iteration ($2 \leq t \leq T$):

$$\begin{aligned} \delta_t(j) &= \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}] \cdot b_j(O_t)\,, \qquad 1 \leq j \leq N\,, \\ \psi_t(j) &= \operatorname*{argmax}_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}]\,, \qquad 1 \leq j \leq N\,. \end{aligned}$$

3. Ende:

$$\begin{aligned} P^* &= \max_{1 \leq i \leq N} \delta_T(i)\,, \\ q_T^* &= \operatorname*{argmax}_{1 \leq i \leq N} \delta_T(i)\,. \end{aligned}$$

4. Backtracking:

$$q_t^* = \psi_{t+1}(q_{t+1}^*)\,, \qquad t = T-1, T-2, \ldots, 1\,.$$

Aus der Analogie zum Forward-Backward-Algorithmus ergibt sich auch hier wieder eine Komplexität von $\mathcal{O}(TN^2)$, da die Rekonstruktion des Viterbi-Pfads nur $\mathcal{O}(N)$ Schritte benötigt.

2.2.3 Baum-Welch-Algorithmus

Das Anpassen der Parameter eines HMM λ an eine gegebene Sequenz O, so dass die Wahrscheinlichkeit $P(O|\lambda)$ maximiert wird, ist das schwierigste der oben formulierten drei Basisprobleme. Im Gegensatz zu den in den vorangehenden Abschnitten betrachteten Fragestellungen ist hierzu kein analytisches Verfahren bekannt. Der im Folgenden beschriebene Baum-Welch-Algorithmus ist ein iteratives Verfahren, bei dem $P(O|\lambda)$ lokal maximiert wird [2, 3], und stellt eine Variante des später formulierten und allgemeineren EM-Algorithmus dar [7], der klassischerweise für „incomplete-data"-Probleme eingesetzt wird [6, 33]. Daneben gibt es auch Ansätze, die klassische Optimierungsmethoden wie z. B. das Gradienten-Verfahren oder die Lagrange-Methode verwenden [28].

Das Prinzip des Baum-Welch-Algorithmus besteht darin, anhand von gegebenen Modellparametern und einer gegebenen Sequenz O neue Schätzer für die gesuchten „optimalen" Parameter zu bestimmen, die die Wahrscheinlichkeit $P(O|\lambda)$ verbessern bzw. nicht kleiner werden lassen. Diese Reestimierungsformeln, die eine einfache Interpretation ermöglichen, werden zunächst intuitiv hergeleitet. Der für spätere Modellerweiterungen wichtige Beweis zur Konvergenz des Iterationsverfahrens wird im anschließenden Abschnitt skizziert. In Abschnitt 2.5 werden die Reestimierungsformeln für den Einsatz von mehreren Sequenzen erweitert.

Wir beginnen mit der Definition der Variablen $\gamma_t(i)$ als der Wahrscheinlichkeit, zur Zeit t im Zustand i zu sein, gegeben die Sequenz O und die Modellparameter λ:

$$\gamma_t(i) := P(q_t = i|O, \lambda) .$$

Mit Hilfe der Forward- und Backward-Variablen und Gleichung (2.6) berechnet sich diese Variable als

$$\begin{aligned} \gamma_t(i) &= \frac{P(q_t = i, O|\lambda)}{P(O|\lambda)} \\ &= \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i)\beta_t(i)} . \end{aligned}$$

Daneben definieren wir eine zweite Variable $\xi_t(i)$ als die Wahrscheinlichkeit eines Zustandswechsels von S_i nach S_j zum Zeitpunkt t, gegeben O und λ:

$$\begin{aligned} \xi_t(i,j) &:= P(q_t = i, q_{t+1} = j|O, \lambda) \\ &= \frac{\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}{P(O|\lambda)} . \end{aligned}$$

Nach der Definition der beiden Variablen gilt dann

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j) .$$

Summieren wir nun $\xi_t(i,j)$ über die Zeit t, dann erhalten wir eine Größe, die der beim Erzeugen der Sequenz O erwarteten Anzahl von Übergängen von Zustand S_i zum Zustand S_j entspricht. Die Summe der $\gamma_t(i)$ über t bestimmt dagegen die erwartete Anzahl von Übergängen aus Zustand S_i, wieder bei Beobachtung der Sequenz O:

$$\begin{aligned} \sum_{t=1}^{T-1} \gamma_t(i) &= \text{erwartete Anzahl der Übergänge aus } S_i \\ \sum_{t=1}^{T-1} \xi_t(i,j) &= \text{erwartete Anzahl der Übergänge von } S_i \text{ nach } S_j \end{aligned}$$

Es liegt nahe, diese Erwartungswerte zur Bestimmung eines neuen Parametersatzes $\bar{\lambda}$ zu benutzen:

$$\begin{aligned} \bar{\pi}_i &= \text{erwartete Wahrscheinlichkeit, zur Zeit } t = 1 \text{ in } S_i \text{ zu sein} \\ &= \gamma_1(i) = \frac{\alpha_1(i)\beta_1(i)}{\sum_{j=1}^{N} \alpha_1(j)\beta_1(j)} , \end{aligned} \tag{2.9a}$$

$$\begin{aligned}
\bar{a}_{ij} &= \frac{\text{erwartete Anzahl der Übergänge } S_i \text{ nach } S_j}{\text{erwartete Anzahl der Übergänge aus } S_i} \\
&= \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} = \frac{\sum_{t=1}^{T-1} \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{t=1}^{T-1} \alpha_t(i)\beta_t(i)}\,, \qquad (2.9b) \\
\bar{b}_{jm} &= \frac{\text{erwartete Anzahl von Ausgaben } v_m \text{ in } S_j}{\text{erwartete Anzahl, in } S_j \text{ zu sein}} \\
&= \frac{\sum_{\substack{t=1 \\ O_t = v_m}}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)} = \frac{\sum_{\substack{t=1 \\ O_t = v_m}}^{T} \alpha_t(i)\beta_t(i)}{\sum_{t=1}^{T} \alpha_t(i)\beta_t(i)}\,. \qquad (2.9c)
\end{aligned}$$

In der Tat kann gezeigt werden, dass sich die Wahrscheinlichkeit $P(O|\lambda)$ durch iteratives Anwenden der obigen Formeln solange verbessert, bis ein kritischer Punkt erreicht ist, der dann ein lokales Maximum darstellt. Der entsprechende Beweis wird im nächsten Abschnitt vorgestellt.

Ein wichtiger Aspekt des Baum-Welch-Algorithmus ist die automatische Einhaltung der stochastischen Nebenbedingungen, denn es gilt zu jeder Zeit:

$$\sum_{j=1}^{N} \bar{\pi}_j = \sum_{j=1}^{N} \bar{a}_{ij} = \sum_{m=1}^{M} \bar{b}_{im} = 1\,, \quad i \in \{1, \ldots, N\}\,.$$

Schießlich bemerken wir noch, dass ein Parameter, der einmal auf null gesetzt wurde, auch im weiteren Verlauf des Algorithmus nicht mehr positiv werden kann, da er jeweils multiplikativ im Nenner der entsprechenden Formel eingeht.

2.2.4 Konvergenz des Baum-Welch-Algorithmus

Basierend auf Baums Beweis [2] für Hidden-Markov-Modelle mit diskreten Ausgaben werden wir zeigen, dass die heuristisch hergeleiteten Reestimierungsformeln die Wahrscheinlichkeit $P(O|\lambda)$ mit jedem Schritt verbessern, bis ein kritischer Punkt, d. h. ein lokales oder sogar das globale Maximum, erreicht wird. Dabei übernehmen wir im Wesentlichen die Darstellung der Beweisskizze in [28].

Zum Konvergenzbeweis verwenden wir folgende zwei Lemmata:

Lemma 2.1 *Seien u_i, $i = 1, \ldots, S$, positive reelle Zahlen, und w_i, $i = 1, \ldots, S$, nichtnegative reelle Zahlen, so dass $\sum_i w_i > 0$. Dann folgt aus der Konkavität der Logarithmus-Funktion:*

$$\begin{aligned}\ln\left(\frac{\sum_i w_i}{\sum_i u_i}\right) &= \ln\Big[\sum_i \frac{u_i}{\sum_j u_j}\cdot\frac{w_i}{u_i}\Big] \\ &\geq \sum_i \frac{u_i}{\sum_j u_j}\ln\big(\frac{w_i}{u_i}\big) \\ &= \frac{1}{\sum_j u_j}\Big[\sum_i (u_i \ln w_i - u_i \ln u_i)\Big]\,. \end{aligned} \tag{2.10}$$

Beweis: Für eine konkave Funktion f und mit $\sum_i \gamma_i = 1$ gilt $f(\sum_i \gamma_i x_i) \geq \sum_i \gamma_i f(x_i)$.

□

Lemma 2.2 *Die Funktion*

$$F(x) = \sum_{i=1}^{n} c_i \ln x_i \tag{2.11}$$

nimmt für $c_i > 0, i = 1, \ldots, n$, unter der Nebenbedingung $\sum_i x_i = 1$ ihr eindeutiges globales Maximum an in $x' = (x'_1, \ldots, x'_n)$ mit

$$x'_i = \frac{c_i}{\sum_k c_k}\,. \tag{2.12}$$

Beweis: $F(x)$ ist konkav, somit existiert ein eindeutiges globales Maximum. Mit der Lagrange-Methode folgt

$$\frac{\partial}{\partial x_i}\Big[F(x) - \rho\sum_i x_i\Big] = \frac{c_i}{x_i} - \rho = 0\,. \tag{2.13}$$

Multiplikation mit x_i und Summation über i ergibt $\rho = \sum_i c_i$ und somit das Ergebnis (2.12).

□

Wir werden nun zunächst eine Hilfsfunktion $\mathcal{Q}(\lambda, \bar{\lambda})$ definieren, von der gezeigt werden kann, dass ihre Maximierung bezüglich $\bar{\lambda}$ einer Verbesserung der Wahrscheinlichkeit $P(O|\bar{\lambda})$ gegenüber $P(O|\lambda)$ entspricht. Schließlich wird sich herausstellen, dass $\mathcal{Q}(\lambda, \bar{\lambda})$ genau dann maximiert wird, wenn $\bar{\lambda} = (\bar{\pi}, \bar{A}, \bar{B})$ wie in den Gleichungen (2.9a) bis (2.9c) gewählt wird.

Für gegebene Modelle λ und $\bar{\lambda}$ sei die sogenannte Q-Funktion definiert als

$$\mathcal{Q}(\lambda, \bar{\lambda}) := \sum_{s=1}^{S} P(Q_s, O|\lambda)\cdot \ln P(Q_s, O|\bar{\lambda})\,, \tag{2.14}$$

wobei S die Anzahl aller möglichen Zustandspfade der Länge T durch die Modelle und Q_s den s-ten Zustandspfad bezeichnen (wir unterscheiden zwischen der Funktion $\mathcal{Q}$ und einer Zustandsfolge Q).

Satz 2.3 *Aus* $\mathcal{Q}(\lambda,\bar{\lambda}) > \mathcal{Q}(\lambda,\lambda)$ *folgt* $P(O|\bar{\lambda}) > P(O|\lambda)$ *und aus* $\mathcal{Q}(\lambda,\bar{\lambda}) \geq \mathcal{Q}(\lambda,\lambda)$ *folgt* $P(O|\bar{\lambda}) \geq P(O|\lambda)$.

Beweis: Sei $\mathcal{Q}(\lambda,\bar{\lambda}) \geq \mathcal{Q}(\lambda,\lambda)$. Wir setzen

$$\begin{aligned} u_s &:= P(Q_s, O|\lambda)\,, \\ w_s &:= P(Q_s, O|\bar{\lambda})\,. \end{aligned} \tag{2.15}$$

Damit gilt

$$\begin{aligned} \sum_{s=1}^{S} u_s &= P(O|\lambda)\,, \\ \sum_{s=1}^{S} w_s &= P(O|\bar{\lambda})\,. \end{aligned} \tag{2.16}$$

Wenn wir in den Summen der Gleichungen (2.14) und (2.16) alle s mit $u_s = 0$ weggelassen, können wir Lemma 2.1 anwenden, und daraus folgt schließlich

$$\ln\left(\frac{P(O|\bar{\lambda})}{P(O|\lambda)}\right) \geq \frac{1}{P(O|\lambda)} \cdot [\mathcal{Q}(\lambda,\bar{\lambda}) - \mathcal{Q}(\lambda,\lambda)] \geq 0\,. \tag{2.17}$$

□

Mit Gleichung (2.1) in Abschnitt 2.2.1 kann der Logarithmus von $P(O|\lambda)$ als Funktion der Modellparameter berechnet werden:

$$\ln P(Q = Q_s, O|\bar{\lambda}) = \ln \bar{\pi}_{q_1} + \sum_{t=1}^{T-1} \ln \bar{a}_{q_t q_{t+1}} + \sum_{t=1}^{T} \ln \bar{b}_{q_t}(O_t)\,. \tag{2.18}$$

Durch Einsetzen von (2.18) in (2.14) erhält man

$$\begin{aligned} \mathcal{Q}(\lambda,\bar{\lambda}) &= \sum_{Q\in\{Q_1,\ldots,Q_S\}} P(Q,O|\lambda) \ln \bar{\pi}_{q_1} + \sum_{Q\in\{Q_1,\ldots,Q_S\}} P(Q,O|\lambda) \sum_{t=1}^{T-1} \ln \bar{a}_{q_t q_{t+1}} \\ &\quad + \sum_{Q\in\{Q_1,\ldots,Q_S\}} P(Q,O|\lambda) \sum_{t=1}^{T} \ln \bar{b}_{q_t}(O_t) \\ &=: \mathcal{Q}_\pi + \mathcal{Q}_a + \mathcal{Q}_b\,. \end{aligned} \tag{2.19}$$

Für die so definierten Teilfunktionen gilt durch Zusammenfassen der entsprechenden Zustands-

pfade:

$$\begin{aligned}
\mathcal{Q}_\pi &= \sum_Q P(Q,O|\lambda)\ln\overline{\pi}_{q_1} \\
&= \sum_{i=1}^{N}\sum_{Q \text{ mit } q_1=i} P(Q,O|\lambda)\ln\overline{\pi}_i \\
&= \sum_{i=1}^{N} P(q_1=i,O|\lambda)\ln\overline{\pi}_i \\
&= \sum_{i=1}^{N} P(O|\lambda)P(q_1=i|O,\lambda)\ln\overline{\pi}_i \\
&= P(O|\lambda)\sum_{i=1}^{N}\gamma_1(i)\ln\overline{\pi}_i\,, \qquad (2.20a)
\end{aligned}$$

$$\begin{aligned}
\mathcal{Q}_a &= \sum_Q P(Q,O|\lambda)\sum_{t=1}^{T-1}\ln\overline{a}_{q_t q_{t+1}} \\
&= \sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{t=1}^{T-1} P(q_t=i,q_{t+1}=j,O|\lambda)\ln\overline{a}_{ij} \\
&= P(O|\lambda)\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{t=1}^{T-1}\xi_t(i,j)\ln\overline{a}_{ij}\,, \qquad (2.20b)
\end{aligned}$$

$$\begin{aligned}
\mathcal{Q}_b &= \sum_Q P(Q,O|\lambda)\sum_{t=1}^{T}\ln\overline{b}_{q_t}(O_t) \\
&= \sum_{j=1}^{N}\sum_{m=1}^{M}\sum_{t=1}^{T} P(q_t=j,O_t=v_m,O|\lambda)\ln\overline{b}_{jm} \\
&= P(O|\lambda)\sum_{j=1}^{N}\sum_{m=1}^{M}\sum_{\substack{t=1\\O_t=v_m}}^{T}\gamma_t(i)\ln\overline{b}_{jm}\,, \qquad (2.20c)
\end{aligned}$$

wobei die Größen $\gamma_t(i)$ und $\xi_t(i,j)$ im vorigen Abschnitt 2.2.3 definiert wurden.

Damit ist die Q-Funktion eine Summe unabhängiger Funktionen des Typs $F(x)$ aus Lemma 2.2 und folglich wird $\mathcal{Q}(\lambda,\overline{\lambda})$ genau dann maximiert, wenn gilt:

$$\overline{\pi}_i = \frac{\gamma_1(i)}{\sum_{j=1}^{N}\gamma_1(j)} = \gamma_1(i)\,, \qquad (2.21a)$$

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{k=1}^{N} \sum_{t=1}^{T-1} \xi_t(i,k)} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}, \tag{2.21b}$$

$$\bar{b}_{jm} = \frac{\sum_{\substack{t=1 \\ O_t=v_m}}^{T} \gamma_t(j)}{\sum_{m=1}^{M} \sum_{\substack{t=1 \\ O_t=v_m}}^{T} \gamma_t(j)} = \frac{\sum_{\substack{t=1 \\ O_t=v_m}}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}. \tag{2.21c}$$

Dies sind aber genau die Reestimierungsformeln (2.9a) – (2.9c), die wir in Abschnitt 2.2.3 intuitiv hergeleitet hatten.

2.3 Hidden-Markov-Modelle mit stetigen Ausgaben

Bei den bisher betrachteten Hidden-Markov-Modellen ist jedem Zustand eine diskrete Verteilungsfunktion über einer gemeinsamen diskreten und endlichen Menge von Ausgabesymbolen zugeordnet. Ein solches Modell kann folglich nur diskrete Sequenzen modellieren bzw. erzeugen. Die Theorie der Hidden-Markov-Modelle kann jedoch erweitert werden zu Modellen mit stetigen Ausgaben, bei denen die Beobachtungen einem d-dimensionalen euklidischen Raum entstammen. In diesem Fall unterliegen die Ausgaben pro Zustand stetigen multivariaten Dichtefunktionen, und statt einer Sequenz von diskreten Symbolen wird eine Folge von d-dimensionalen Beobachtungsvektoren erzeugt. Abbildung 2.4 zeigt ein einfaches Beispiel eines HMM mit stetigen, eindimensionalen Ausgaben ($d = 1$).

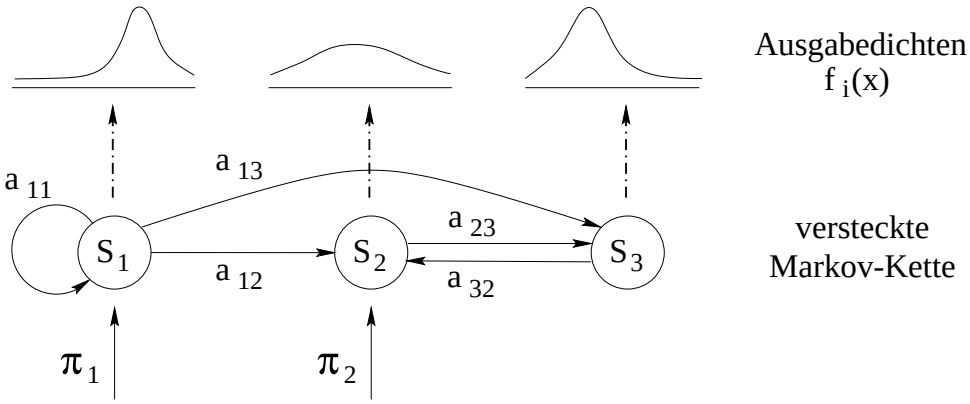


Abbildung 2.4: Beispiel für ein einfaches HMM mit stetigen Ausgaben

Die verwendete Klasse von Dichtefunktionen muss bestimmte Anforderungen erfüllen, wenn zum Training der Modelle entsprechende Reestimierungsformeln für die Parameter der Dichtefunktionen aufgestellt werden sollen. Die allgemeinste Form einer solchen Funktion, für die

in der Literatur eine Reestimierungs-Methode formuliert wurde, besteht aus einer Mischverteilung von entweder log-konkaven oder elliptisch-symmetrischen Dichten [21]. Zu dem ersten Fall zählen alle Funktionen $f(x)$, für die $\log f(x)$ streng konkav in x ist und die in $x = 0$ ein eindeutiges Maximum haben, sowie davon abgeleitete Funktionen der Form $\sigma^{-1} f((x-\mu)/\sigma)$ mit beliebigen Skalierungsparametern μ und $\sigma > 0$ [2,3]. Eine elliptisch-symmetrisch Funktion über dem d-dimensionalen Vektor x ist definiert als Funktion einer positiv-definiten quadratischen Form:

$$|\Sigma|^{-1/2} f(q(x)), \quad \text{mit } q(x) = (x-m)^T \Sigma^{-1} (x-m) .$$

Die $d \times d$-Skalierungsmatrix Σ muss dabei positiv-definit und symmetrisch sein, während der Lagevektor m einen beliebigen Punkt im d-dimensionalen euklidischen Raum darstellt [29].

In der Praxis werden als Mischkomponenten meist Wahrscheinlichkeitsdichten der (multivariaten) Normalverteilung eingesetzt, da mit einer solchen Mischverteilung im Prinzip jede stetige Verteilung approximiert werden kann. Die Dichtefunktion eines Ausgabevektors x auf dem Zustand j schreibt sich dann folgendermaßen:

$$f_j(x) = \sum_{m=1}^{M} c_{jm} \mathcal{N}(x, \mu_{jm}, U_{jm}) = \sum_{m=1}^{M} c_{jm} f_{jm}(x) , \tag{2.22}$$

wobei c_{jm} den Koeffizienten, μ_{jm} den Erwartungswertvektor und U_{jm} die Kovarianzmatrix der m-ten Mischkomponente in Zustand j beschreibt. Die c_{jm} gehorchen den stochastischen Nebenbedingungen

$$\begin{aligned} \sum_{m=1}^{M} c_{jm} &= 1 , \qquad 1 \le j \le N , \\ c_{jm} &\ge 0 , \qquad 1 \le j \le N , \; 1 \le m \le M . \end{aligned}$$

Für das HMM mit stetigen Ausgaben wird analog zum diskreten Modell und mit $C = \{c_{jm}\}$, $\mu = \{\mu_{jm}\}$ und $U = \{U_{jm}\}$ die Kurzschreibweise $\lambda := (\pi, A, C, \mu, U)$ verwendet. Statt der Wahrscheinlichkeit $P(O|\lambda)$, die für ein solches Modell aufgrund der stetigen Verteilungen gleich null ist, betrachten wir die Dichte einer beobachteten Sequenz $O = O_1 \cdots O_T$:

$$L(O|\lambda) := \sum_{\text{alle } Q} \pi_{q_1} f_{q_1}(O_1) \prod_{t=2}^{T} a_{q_{t-1} q_t} f_{q_t}(O_t) . \tag{2.23}$$

Darin bezeichne $Q = q_1 \cdots q_T$ wieder eine Zustandsfolge der Länge T. Es ist zu beachten, dass die Elemente der Sequenz O im Allgemeinen Vektoren sind. Die Funktion $L(O|\lambda)$ bezeichnen wir im Weiteren als die Likelihood der Sequenz O, gegeben Modell λ.

Die Berechnung sowohl der Forward- und Backward-Variablen als auch des Viterbipfades kann nach den Abschnitten 2.2.1 und 2.2.2 erfolgen, indem überall die diskrete Funktion b_j gegen die Dichte f_j ausgetauscht wird und alle Wahrscheinlichkeiten als Dichten interpretiert werden. Mit

Hilfe der so angepassten Forward- und Backward-Variablen definieren wir folgende Dichtefunktionen:

$$\begin{aligned}
\gamma_t(i) &= L(q_t = i|O,\lambda) \\
&= \frac{L(q_t = i, O|\lambda)}{L(O|\lambda)} \\
&= \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{N}\alpha_t(i)\beta_t(i)}\,, \qquad (2.24a)\\
\xi_t(i,j) &= L(q_t = i, q_{t+1} = j|O,\lambda) \\
&= \frac{L(q_t = i, q_{t+1} = j, O|\lambda)}{L(O|\lambda)} \\
&= \frac{\alpha_t(i)\, a_{ij}\, f_j(O_{t+1})\beta_{t+1}(j)}{\sum_{i=1}^{N}\alpha_t(i)\beta_t(i)}\,, \qquad (2.24b)\\
\zeta_t(j,m) &= L(q_t = j, m|O,\lambda) \\
&= \frac{L(q_t = j, m, O|\lambda)}{L(O|\lambda)} \\
&= \begin{cases} \dfrac{\sum_{i=1}^{N}\alpha_{t-1}(i)\, a_{ij}\, c_{jm}\, f_{jm}(O_t)\beta_t(j)}{\sum_{i=1}^{N}\alpha_t(i)\beta_t(i)} & t > 1 \\ \dfrac{\pi_j\, c_{jm}\, f_{jm}(O_t)\beta_t(j)}{\sum_{i=1}^{N}\alpha_t(i)\beta_t(i)} & t = 1\,. \end{cases} \qquad (2.24c)
\end{aligned}$$

Damit sind die $\gamma_t(i)$ und $\xi_t(i,j)$ die stetigen Varianten der gleichnamigen Variablen in Abschnitt 2.2.3. Die neu eingeführte Größe $\zeta_t(j,m)$ bezeichnet die gemeinsame Dichte eines Zustands i und der Mischkomponente m zum Zeitpunkt t.

Mit diesen Größen können wiederum Reestimierungsformeln aufgestellt werden, mit denen die Likelihood $L(O|\lambda)$ iterativ maximiert wird [20]:

$$\begin{aligned}
\bar{\pi}_i &= \gamma_1(i)\,, \qquad (2.25a)\\
\bar{a}_{ij} &= \frac{\sum_{t=1}^{T-1}\xi_t(i,j)}{\sum_{t=1}^{T-1}\gamma_t(i)}\,, \qquad (2.25b)\\
\bar{c}_{jm} &= \frac{\sum_{t=1}^{T}\zeta_t(j,m)}{\sum_{t=1}^{T}\gamma_t(j)}\,, \qquad (2.25c)\\
\bar{\mu}_{jm} &= \frac{\sum_{t=1}^{T}\zeta_t(j,m)O_t}{\sum_{t=1}^{T}\zeta_t(j,m)}\,, \qquad (2.25d)\\
\bar{U}_{jm} &= \frac{\sum_{t=1}^{T}\zeta_t(j,m)(O_t-\bar{\mu}_{jm})(O_t-\bar{\mu}_{jm})'}{\sum_{t=1}^{T}\zeta_t(j,m)}\,. \qquad (2.25e)
\end{aligned}$$

Die Gleichungen für die $\overline{\pi}_i$ und $\overline{a}_{ij}$ entsprechen den Gleichungen (2.9a) und (2.9b), wieder bei Austausch der diskreten Dichtefunktion gegen die stetige Dichte der Mischverteilung. Mit $(\cdot)'$ in der letzten Gleichung wird der transponierte Vektor bezeichnet.

Die obigen Reestimierungsformeln lassen wieder eine anschauliche Interpretation analog zu den Formeln für das HMM mit diskreten Ausgaben zu. Der Konvergenzbeweis des Iterationsverfahrens mit diesen Formeln basiert ebenso wie im diskreten Fall auf der Maximierung einer Hilfsfunktion Q und und verläuft entsprechend ähnlich zu dem Beweis in Abschnitt 2.2.4. Ein guter Überblick dazu findet sich in [17], für detailliertere Ausführungen siehe [2, 3, 20, 29].

Abschließend folgt noch eine Bemerkung zum Erzeugen von Zufallszahlen einer Mischverteilung: Wenn wir zuerst mittels der Wahrscheinlichkeiten c_m eine Mischkomponente m auswürfeln und danach eine Zufallszahl X über der Dichte f_m ermitteln, dann sind die Werte dieser Zufallsvariablen genau nach einer Mischverteilung mit Dichtefunktion $\sum_m c_m f_m$ verteilt ([23], S. 58 f).

2.4 Skalierung

Bei der Umsetzung der Basisalgorithmen in Computerprogramme stellt sich heraus, dass die Forward- und Backward-Variablen im Laufe der rekursiven Berechnungen sehr schnell den darstellbaren Bereich der Fließkommazahlen des Rechners verlassen. Dies ist nicht verwunderlich, wenn wir die Definition der $\alpha_t(i)$ in (2.2) betrachten; in der Tat strebt $\alpha_T(i)$ als Produkt der im Allgemeinen sehr kleinen a_{ij} und $b_j(O_t)$ für $T \to \infty$ exponentiell gegen null. Die Lösung dieses Problems liegt darin, die $\alpha_t(i)$ in jedem Schritt der rekursiven Berechnung so zu skalieren, dass die Variablen immer im darstellbaren Bereich des Computers liegen.

Eine detaillierte Beschreibung der Skalierungsmethode findet sich in [43], das eine Ergänzung zu [38] bzw. [28] darstellt. Da im Laufe dieser Arbeit im Zuge von Modellerweiterungen nochmals auf die Skalierung eingegangen wird, soll das Verfahren hier etwas ausführlicher skizziert werden.

Zunächst erweitern wir den Forward-Algorithmus derart, dass wir in jedem Zeitschritt zusätzliche, normierte Variablen berechnen:

1. $t = 1$:

$$\begin{aligned} \tilde{\alpha}_1(i) &:= \alpha_1(i) = \pi_i b_i(O_1), \qquad 1 \le i \le N, \\ c_1 &:= \frac{1}{\sum_{i=1}^{N} \tilde{\alpha}_1(i)}, \\ \hat{\alpha}_1(i) &:= c_1 \cdot \tilde{\alpha}_1(i). \end{aligned} \tag{2.26}$$

2. $t = 2, \ldots, T$:

$$\tilde{\alpha}_t(i) := \left[\sum_{j=1}^{N} \hat{\alpha}_{t-1}(j) a_{ji} \right] b_i(O_t), \quad 1 \le i \le N, \tag{2.27a}$$

$$c_t := \frac{1}{\sum_{i=1}^{N} \tilde{\alpha}_t(i)}, \tag{2.27b}$$

$$\hat{\alpha}_t(i) \;:=\; c_t \cdot \tilde{\alpha}_t(i)\,. \tag{2.27c}$$

Per Induktion lässt sich zeigen, dass gilt:

$$\hat{\alpha}_t(i) = \Big(\prod_{\tau=1}^{t} c_\tau\Big)\alpha_t(i)\,, \tag{2.28}$$

und (2.27b), (2.27a) und (2.28) eingesetzt in (2.27c) führen zu

$$\hat{\alpha}_t(i) = \frac{\alpha_t(i)}{\sum_{j=1}^{N}\alpha_t(j)}\,. \tag{2.29}$$

Wie mit (2.29) zu sehen ist, wird damit jedes $\alpha_t(i)$ mit der Summe der $\alpha_t(i)$ über alle Zustände skaliert, d. h. $\hat{\alpha}_t(i)$ entspricht der bedingten Wahrscheinlichkeit $P(q_t = i|O_1 \cdots O_t, \lambda)$ (vgl. Gleichung (2.2)).

Die Backward-Variablen $\beta_t(i)$ werden jetzt mit den gleichen Skalierungsfaktoren wie die $\alpha_t(i)$ multipliziert, und in ähnlicher Weise ergeben sich

$$\begin{aligned}
\hat{\beta}_T(i) &:= \beta_T(i) = 1\,,\\
\tilde{\beta}_T(i) &:= c_T\hat{\beta}_T(i)\,,\\
\hat{\beta}_t(i) &:= \sum_{j=1}^{N} a_{ij}b_j(O_{t+1})\tilde{\beta}_{t+1}(j) = \Big(\prod_{\tau=t+1}^{T} c_\tau\Big)\beta_t(i)\,,\\
\tilde{\beta}_t(i) &:= c_t\hat{\beta}_t(i)\,.
\end{aligned} \tag{2.30}$$

Damit kann z. B. die Reestimierungsformel (2.9b) für die Übergangswahrscheinlichkeiten $\bar{a}_{ij}$ als Funktion der skalierten Forward- und Backward-Variablen formuliert werden, denn mit $C_t := \prod_{\tau=1}^{t} c_\tau$ und $D_{t+1} := \prod_{\tau=t+1}^{T} c_\tau$ ist $C_tD_{t+1} = C_T$ unabhängig von t, und es gilt (mit $D_{T+1} := 1$)

$$\begin{aligned}
\bar{a}_{ij} &= \frac{\sum_{t=1}^{T-1}\alpha_t(i)\,a_{ij}\,b_j(O_{t+1})\,\beta_{t+1}(j)}{\sum_{t=1}^{T-1}\alpha_t(i)\,\beta_t(i)}\\
&= \frac{\sum_{t=1}^{T-1}C_T\,\alpha_t(i)\,a_{ij}\,b_j(O_{t+1})\,\beta_{t+1}(j)}{\sum_{t=1}^{T-1}C_T\,\alpha_t(i)\,\beta_t(i)}\\
&= \frac{\sum_{t=1}^{T-1}C_t\alpha_t(i)\,a_{ij}\,b_j(O_{t+1})\,D_{t+2}\beta_{t+1}(j)\,c_{t+1}}{\sum_{t=1}^{T-1}C_t\alpha_t(i)\,D_{t+1}\beta_t(i)}\\
&= \frac{\sum_{t=1}^{T-1}\hat{\alpha}_t(i)\,a_{ij}\,b_j(O_{t+1})\,\hat{\beta}_{t+1}(j)\,c_{t+1}}{\sum_{t=1}^{T-1}\hat{\alpha}_t(i)\,\hat{\beta}_t(i)}\,.
\end{aligned} \tag{2.31}$$

Es liegt auf der Hand, dass die anderen Reestimierungskoeffizienten auf die gleiche Art angepasst werden können (alle skalierten Formeln sind im Anhang A aufgelistet).

Während sich also die Reestimierungsformeln nur unwesentlich ändern, müssen wir feststellen, dass die Gleichung $P(O|\lambda) = \sum_{i=1}^{N}\alpha_T(i)$ zur Bestimmung der Likelihood in dieser Form mit den

skalierten Größen nicht mehr verwendet werden kann. Allerdings gilt jetzt

$$\begin{aligned}\Big(\prod_{\tau=1}^{T} c_\tau\Big) P(O|\lambda) &= \Big(\prod_{\tau=1}^{T} c_\tau\Big) \sum_{i=1}^{N} \alpha_T(i) \\ &= \sum_{i=1}^{N} \Big(\prod_{\tau=1}^{T} c_\tau\Big) \alpha_T(i) \\ &= \sum_{i=1}^{N} \hat{\alpha}_T(i) \\ &= 1\,. \end{aligned} \tag{2.32}$$

Damit kann der Logarithmus der Likelihood, der im darstellbaren Bereich des Computers liegt, folgendermaßen berechnet werden:

$$\log P(O|\lambda) = -\sum_{\tau=1}^{T} \log c_\tau\,. \tag{2.33}$$

Schließlich kann auch der Viterbi-Algorithmus so angepasst werden, dass die logarithmierte Likelihood $\log P(O|\lambda)$ direkt berechnet wird und alle Größen im darstellbaren Bereich des Rechners liegen. Dazu definieren wir statt (2.7) die Variable

$$\Phi_t(i) := \max_{q_1,q_2,\ldots q_t} [\log P(q_1 q_2 \cdots q_t = i, O_1 O_2 \cdots O_t|\lambda)] \tag{2.34}$$

und berechnen rekursiv

$$\Phi_t(j) = \max_{1\le i\le N} [\Phi_{t-1}(i) + \log a_{ij}] + \log b_j(O_t)\,, \tag{2.35}$$

so dass am Ende $\log P^* = \max_{1\le i\le N} [\Phi_T(i)]$ gilt.

2.5 Reestimierung mit mehreren Sequenzen

Bei vielen praktischen Anwendungen geht es darum, mehrere relativ kurze Sequenzen mit einem Hidden-Markov-Modell abzubilden: In der Spracherkennung sollen z. B. mit einem Modell verschiedene Aussprachen eines Wortes abgebildet werden, und in den Anwendungen dieser Arbeit möchten wir viele kurze Zeitreihen durch wenige Modelle klassifizieren. In solchen Fällen macht es wenig Sinn, ein HMM nur mit einer einzigen Sequenz zu trainieren. Die Reestimierungsformeln (2.9a) – (2.9c) lassen sich jedoch problemlos auf das Training mit mehreren Sequenzen erweitern [17, 28].

Wir setzen voraus, dass die K gegebenen Sequenzen $O^{(1)}, \ldots, O^{(K)}$ unabhängig voneinander sind, und bezeichnen mit $O^{(k)} = O_1^{(k)} \cdots O_{T^k}^{(k)}$ die k-te Sequenz der individuellen Länge T^k. Das Training mit diesen Sequenzen basiert dann auf der Maximierung von

$$P(O^{(1)}, \ldots, O^{(K)}|\lambda) = \prod_{k=1}^{K} P(O^{(k)}|\lambda)\,.$$

Intuitiv ergeben sich angepasste Parametergleichungen mit folgender Überlegung: Sei $\sum_{t=1}^{T^k-1} \xi_t^k(i,j)$ die erwartete Anzahl der Übergänge von Zustand S_i nach S_j bei Ausgabe der Sequenz $O^{(k)}$. Dann erhalten wir durch Summation über k den Erwartungswert der Anzahl dieser Übergänge bezüglich aller Sequenzen. Analog zur Gleichung (2.9b) lässt sich ein Schätzer der Übergangswahrscheinlichkeit a_{ij} bei Beobachtung mehrerer Sequenzen bestimmen als

$$\bar{a}_{ij} = \frac{\sum_{k=1}^{K} \sum_{t=1}^{T^k-1} \xi_t^k(i,j)}{\sum_{k=1}^{K} \sum_{t=1}^{T^k-1} \gamma_t^k(i)} \,. \tag{2.36}$$

Die Berechnung der anderen Parameter kann auf die gleiche Art und Weise angepasst werden, indem in den Gleichungen (2.9a) und (2.9c) im Zähler und Nenner zusätzlich über k summiert wird.

Auch der Konvergenzbeweis der Reestimierungsformeln für eine Sequenz in Abschnitt 2.2.4 kann leicht auf den Fall mehrerer Sequenzen übertragen werden. Dazu wird die Q-Funktion definiert als

$$\mathcal{Q}(\lambda, \bar{\lambda}) = \sum_{k=1}^{K} \frac{1}{P(O^{(k)}|\lambda)} \cdot \mathcal{Q}^k(\lambda, \bar{\lambda}) \,,$$

wobei $\mathcal{Q}^k(\lambda, \bar{\lambda})$ die in (2.14) definierte Q-Funktion der k-ten Sequenz ist. Wieder mit Lemma 2.1 lässt sich zeigen, dass aus der Maximierung von $\mathcal{Q}(\lambda, \bar{\lambda})$ eine Maximierung von $\prod_{k=1}^{K} P(O^{(k)}|\lambda)$ folgt. Ein Ausschreiben der Q-Funktion führt dann wieder zu einer Zerlegung in unabhängige Summanden für die einzelnen Parameter. In diesem Fall entspricht jeder dieser Terme einer Funktion des Typs $\sum_k \sum_i c_{ki} x_i$. Unter der gleichen Nebenbedingung $\sum_i x_i = 1$ wie in Lemma 2.2 nimmt solch eine Funktion ihr Maximum in $x_i = \sum_k c_{ki} / \sum_k \sum_j c_{kj}$ an. Damit wird die Q-Funktion maximiert, wenn die $\bar{a}_{ij}$ nach Gleichung (2.36) und analog dazu die Parameter $\bar{\pi}_i$ und $\bar{b}_{jm}$ wie oben beschrieben bestimmt werden.

Wenn wir jetzt die Gleichung (2.36) als Funktion der Forward- und Backward-Variablen und der Modellparameter ausschreiben, fällt auf, dass sich der Faktor $1/P(O^{(k)}|\lambda)$ in Zähler und Nenner nicht mehr wegkürzt wie in Gleichung (2.9b), da er von k abhängt. Dieses Problem löst sich aber von selbst, wenn wir mit den skalierten Forward- und Backward-Variablen rechnen. Denn zur Skalierung hatten wir die ursprüngliche Gleichung (2.9b) in (2.31) um den Faktor $C_T = \prod_{\tau=1}^{T} c_\tau$ erweitert, und mit (2.32) gilt:

$$C_T = 1/P(O|\lambda) \,.$$

Somit erhalten wir aus den skalierten Reestimierungsformeln durch zusätzliche Summation über k in Zähler und Nenner die entsprechenden skalierten Formeln für das Trainieren eines Modells mit mehreren Sequenzen. Im Anhang A sind diese Reestimierungsgleichungen für alle Parameter von Modellen mit diskreten und mit stetigen Ausgaben aufgelistet.

2.6 Praktische Fragen, Modifikationen und Erweiterungen

Die in diesem Kapitel vorgestellten Algorithmen und Methoden stellen das Grundgerüst für die Verwendung von Hidden-Markov-Modellen dar. Darüber hinaus ergeben sich für den praktischen Einsatz der Modelle weitere Aspekte, zusätzliche Fragen und daraus resultierende mögliche Modifikationen und Erweiterungen für die Hidden-Markov-Modellierung, von denen wir hier einige kurz ansprechen und auf die wir teilweise in späteren Kapiteln noch näher eingehen werden.

Modelltopologie

Die Frage nach der Wahl der „richtigen“ Topologie des Modellgraphen (vgl. Abschnitt 2.1) für die jeweilige Anwendung lässt sich nicht einfach beantworten. Oft ergibt sich aus der Problemstellung heraus eine ungefähre Vorstellung für eine passende HMM-Struktur; das für das individuelle Problem geeignete Modell kann dann durch Ausprobieren und Vergleichen verschiedener Alternativen bestimmt werden. Diesem Prinzip sind wir auch bei den Anwendungen in dieser Arbeit gefolgt (siehe Abschnitte 3.3.3 und 7.2).

In der Literatur finden sich auch Ansätze, die Modellstruktur systematisch zu optimieren. In [41] z. B. wird ein Algorithmus vorgeschlagen, der mit einem HMM startet, das die Trainingsdaten mit der größtmöglichen Likelihood abbildet, indem es für jede Ausgabe jeder Sequenz einen eigenen Zustand bereitstellt. Dieses komplexe Modell wird in den nachfolgenden Iterationen durch Zusammenlegen von Zuständen und Kanten nach bestimmten Regeln immer weiter vereinfacht. Als Abbruchkriterium dient die maximale A-posteriori-Likelihood, die sich aus der kleiner werdenden Likelihood und für einfachere Modelle größer werdenden A-priori-Wahrscheinlichkeiten zusammensetzt.

Initialisierung, Sensitivität und Modellvergleich

Aufgrund der nur lokalen Konvergenz des Baum-Welch-Algorithmus beeinflussen die Initialisierungen der Modelle die trainierten Modellparameter. In [8] wird gezeigt, dass bereits für kleine Modelle mit wenigen diskreten Ausgaben eine starke Abhängigkeit der trainierten Parameter von den jeweiligen Initialisierungen auftreten kann. Mit Hilfe des in [22] vorgestellten Abstandsmaßes zwischen Hidden-Markov-Modellen werden in der gleichen Arbeit [22] Sensitivitätsanalysen bei Modellen mit diskreten Ausgaben durchgeführt. Es zeigt sich dort, dass die Matrix B der diskreten Ausgabewahrscheinlichkeiten stärker auf Änderungen der Initialisierungsparameter reagiert als die Matrix A. In [39] dagegen werden Modelle mit Mischungen von stetigen Normalverteilungen betrachtet. Dort wird eine besonders starke Empfindlichkeit der Mittelwerte gegenüber „schlechter“ Initialisierung beobachtet, während sich die Startwerte der anderen Parameter viel weniger stark auf das Modelltraining auswirken. Dabei dient wieder das Abstandsmaß aus [22] als Grundlage für die Modellvergleiche.

In [30] wird ein weiteres Abstandsmaß für Hidden-Markov-Modelle mit diskreten Ausgaben vorgeschlagen. Auf beide Maße werden wir in Abschnitt 4.3.2 etwas genauer eingehen.

Modifikationen

Neben dem in Abschnitt 2.2.3 vorgestellten Baum-Welch-Algorithmus existieren verschiedene andere Verfahren zum Trainieren der Parameter eines HMM. In [1] werden z. B. mehrere Variationen eines Lernalgorithmus vorgeschlagen, die sich alle als Gradientenabstiegsverfahren bei verschiedenen, auf der Likelihood $L(O|\lambda)$ basierenden Zielfunktionen interpretieren lassen. Der Vorteil dieser Algorithmen besteht darin, dass sie im Gegensatz zum Baum-Welch-Algorithmus auch als Online-Verfahren eingesetzt werden können, bei denen ein HMM nacheinander mit einzelnen Sequenzen (nach-)trainiert wird. Ein Verfahren, das sich auch zu einem Modelltraining mit nur wenigen Daten einsetzen lässt, stellt das MAP-Training dar (maximum a posteriori estimation, [4, 13]). Hier wird ähnlich wie bei oben erwähnten Verfahren zum Bestimmen einer optimalen Modelltopologie bei der Maximierung der Likelihood zusätzlich eine a-priori-Wahrscheinlichkeit für jedes Modell berücksichtigt.

Weitere Modifikationen, wie sie in [38] kurz beschrieben werden (siehe Literaturverweise dort), bestehen z. B. in einer Koppelung von Parametern, so dass beispielsweise bestimmte Zustände den gleichen Ausgabeparametern unterliegen, dem Zulassen von Übergangswahrscheinlichkeiten, die keine Ausgaben auf den Zuständen hervorrufen (null transitions) oder in der Abbildung einer expliziten Verweildauer des Markov-Prozesses in einem Zustand, die über eine vorgegeben Dichtefunktion gesteuert wird.

Die hier erwähnten Modifikationen wurden in unserer Arbeit aus verschiedenen Gründen nicht weiter berücksichtigt. So können wir z. B. auf sehr viele Daten zurückgreifen und sehen keinen Bedarf zur Verwendung von Online-Verfahren. Der Einsatz von expliziten Verweildauer-Dichten für die Modellzustände führt dagegen zu einer erhöhten Komplexität des Trainingsalgorithmus.

Modellerweiterungen

Bei dem bisher betrachteten HMM durchläuft der stochastische Prozess eine versteckte Markov-Kette mit endlich vielen, diskreten Zuständen, an denen jeweils eine Ausgabe erzeugt wird. In [14] wird als eine mögliche Verallgemeinerung dieses Konzeptes das sogenannte *factorial HMM* vorgestellt. Bei diesem Modell kann jedes beobachtete Ausgabe-Symbol von mehreren gekoppelten Markov-Ketten abhängen. Das exakte Trainieren der Modellparameter ist bei dieser Modellklasse nicht mehr handhabbar; mittels Monte-Carlo-Verfahren lassen sich die Parameter jedoch in effizienter Zeit ausreichend gut approximieren. Eine andere Verallgemeinerung besteht darin, die diskreten Zustände auf einen stetigen Zustandsraum zu erweitern. Dies führt zu einem *Monte-Carlo-HMM* (MCHMM), bei dem die Zustandsübergänge zum (weiterhin diskreten) Zeitpunkt t durch stetige bedingte Wahrscheinlichkeitsverteilungen beschrieben werden und dessen Modellparameter mit Hilfe einer Monte-Carlo-Version des Baum-Welch-Algorithmus trainiert werden können [42].

In Kapitel 6 dieser Arbeit werden wir ein erweitertes HMM vorstellen, bei dem der Übergang zwischen je zwei Zuständen zum Zeitpunkt t von den bis dahin generierten Ausgaben abhängt und über mehrere diskrete Übergangswahrscheinlichkeiten abgebildet wird.

Kapitel 3

Modellierung von Zeitreihen eines Bausparkollektivs

Das Verhalten der Sparer eines Bausparkollektivs spiegelt sich in Zeitreihen wider, die teilweise stark stochastisch und teilweise durch die Rahmenbedingungen des Bausparvertrags determiniert sind. Die Erwartungen an die künftige Entwicklung dieser Zeitreihen bzw. des in ihnen kodierten Sparerverhaltens bildet die Grundlage jeglicher Planungen bei den Bausparkassen, wobei die genaue Kenntnis des aktuellen Verhaltens die notwendige Basis für jede Simulation darstellt. Eine systematische Datenanalyse und -klassifizierung dient zudem als Grundlage für die Entwicklung von neuen Produkten und Marketingstrategien.

Im Rahmen einer langjährigen Zusammenarbeit der Arbeitsgruppe am ZPR/ZAIK mit den Landesbausparkassen wurden in den letzten Jahren verschiedene Modelle zur Simulation von Bausparkollektiven entwickelt und eingesetzt. Daher steht uns auch ein großer und detaillierter Datenbestand von Einzelverträgen verschiedener Bausparkassen zur Verfügung, der bis zu 15 Jahre zurückreicht.

In diesem Kapitel erläutern wir zunächst den Ablauf eines Bausparvertrags und die Funktionsweise des kollektiven Bausparens, um dann kurz auf die Simulationsmodelle einzugehen, die bisher am ZPR/ZAIK entwickelt wurden. Danach werden wir den Einsatz von Hidden-Markov-Modellen zur Modellierung und Simulation von Zeitreihen eines Bausparkollektivs diskutieren sowie Anforderungen an solche Modelle formulieren, die sich aus der speziellen Anwendung ergeben.

3.1 Das Prinzip des Bausparens

Wir möchten das Thema Bausparen hier nur soweit behandeln, wie es für das Verständnis der weiteren Arbeit nötig ist. Eine ausführliche Darstellung des Ablaufs eines Bausparvertrags und der gesetzlichen Grundlagen findet sich z. B. in [5, 26] und [44].

Der Grundgedanke des Bausparens besteht darin, dass Bauwillige, die eine bestimmte Geldsumme zum Bauen benötigen, eine Zeit lang ihre Sparleistungen in einen gemeinsamen Topf

einzahlen, aus dem nach Erbringung bestimmter Leistungen der jeweilige Rest des erforderlichen Finanzierungsbedarfs als günstiges Darlehen gewährt wird. Die Bausparkasse verwaltet als Wirtschaftsunternehmen die beteiligten Bausparverträge, die jeweils in einem bestimmten Tarif abgeschlossen werden, der den Verlauf des Vertrags in vielen Punkten steuert. Die Gesamtheit der Verträge einer Bausparkasse wird als Bausparkollektiv bezeichnet.

Ein einzelner Bausparvertrag wird i. d. R. über eine bestimmte Bausparsumme (BS) abgeschlossen, die der benötigten Finanzierungssumme entspricht. In der ersten Phase des Vertrags, der sogenannten Sparphase, wird der Vertrag solange bespart, bis die Bedingungen für das Anrecht auf ein Darlehen erreicht sind. Diese Zuteilungsbedingungen bestehen üblicherweise darin, dass der Anspargrad, also das angesparte Guthaben bezogen auf die Bausparsumme, je nach Tarif mindestens 40% oder 50% betragen muß und die Bewertungszahl, kurz BWZ, sowohl eine vertraglich vorgegebene Mindest-Bewertungszahl als auch die aktuelle Ziel-Bewertungszahl überschreiten muß. Die BWZ bewertet die Sparleistung, die der Sparer für das Kollektiv erbracht hat, und errechnet sich bei unseren Daten zum Zeitpunkt t folgendermaßen (Kapitalzinsverfahren):

$$\mathrm{BWZ}(t) = \frac{\alpha * \text{Guthaben}(t) + \beta * \text{summierte Zinsen}(t)}{\gamma * \text{Bausparsumme}(t)} .$$

Die durch den Tarif festgelegten Faktoren α, β und γ dienen dabei einer Gewichtung der einzelnen Größen. Auf die Ziel-Bewertungszahl werden wir gleich noch eingehen.

Nach Erreichen der Zuteilungsbedingungen, deren Erfüllung an bestimmten festen Terminen im Jahr, den Bewertungsstichtagen, überprüft wird, wird der Vertrag zugeteilt; d. h. der Sparer hat jetzt Anspruch auf die Auszahlung seines Guthabens und des Darlehens, also der gesamten Bausparsumme. In dieser Zuteilungsphase gibt es für den Bausparer verschiedene Möglichkeiten: Das Guthaben und das Darlehen können in mehreren Stufen und zeitlich verzögert ausgezahlt werden, und auf das Darlehen kann teilweise oder ganz verzichtet werden.

Mit der ersten Auszahlung des Darlehens tritt der Vertrag in die Darlehens- oder auch Tilgungsphase ein. In diesem Stadium hat der Bausparer wenig Freiheiten, da das Darlehen mit tariflich festgelegter Tilgungsrate und festen Darlehenszinsen zurückgezahlt werden muss. In Absprache mit der Bausparkasse können allerdings Sonderzahlungen geleistet werden, um die Vertragslaufzeit zu verkürzen.

Von besonderem Interesse für die Modellierung von Bausparkollektiven ist die Sparphase, da in diesem Abschnitt des Bausparvertrags der Sparer in seinen Handlungen weitgehend frei ist und individuelle Sparziele verfolgen kann. Neben der nicht vorgeschriebenen Höhe der Sparzahlungen zu beliebigen Zeitpunkten – die Regelsparrate des Tarifs stellt nur einen Richtwert dar – gibt es noch weitere Möglichkeiten, den Verlauf eines Bausparvertrags zu beeinflussen: Erhöhung oder Erniedrigung der Bausparsumme, Tarifwechsel, Aufteilung auf mehrere Verträge, Zusammenlegung von Verträgen oder Kündigung, um nur die wichtigsten zu nennen. Ein zuteilungsreifer Vertrag kann auch fortgesetzt werden, d. h. der Sparer verzichtet vorläufig auf die Zuteilung und bleibt solange in der Sparphase, bis er sein Guthaben und eventuell das Darlehen in Anspruch nehmen will.

Die Kenngrößen eines Bausparkollektivs lassen sich alle aus den summierten bzw. den davon abgeleiteten Größen der Einzelverträge berechnen. Durch die oben beschriebene Vielzahl von

kombinierbaren Verhaltensmöglichkeiten werden eine Analyse und eine Modellierung des Kollektivs erschwert. In gewisser Weise sind die Bausparverträge eines Kollektivs auch miteinander gekoppelt, da die Auszahlung der Verträge bzw. ihr Zuteilungszeitpunkt von der Zuteilungsmasse (angesammelte Bausparmittel, in die im Wesentlichen die Spar- und die Tilgungsleistungen einfließen) abhängt. Anhand dieser Zuteilungsmasse wird die Ziel-Bewertungszahl festgesetzt, die ein Vertrag erreichen muss, um an einem bestimmten Zuteilungstermin noch berücksichtigt zu werden, und mit der die Bausparkasse die Zuteilungen in einem gewissen Rahmen steuern kann. Allerdings spielt diese Koppelung der Verträge in der Praxis durch nur kleine Schwankungen der Zuteilungsmasse eine geringe Rolle und wird in den Simulationsmodellen meist dadurch abgebildet, dass die Ziel-Bewertungszahlen für den ganzen Simulatonszeitraum vorgegeben werden. Durch Variation dieser Zahlen können auch große prognostizierte Änderungen im Neugeschäft, die sich stark auf die Zuteilungsmasse auswirken, simuliert werden.

3.2 Bisherige Modellansätze

Seit den achtziger Jahren wurden in unserer Arbeitsgruppe verschiedene Modelle zur Simulation von Bausparkollektiven entwickelt und in der Praxis auch kontinuierlich eingesetzt. Dabei wurde versucht, mit jedem Modell den jeweiligen aktuellen Bedürfnissen und Fragestellungen gerecht zu werden und gleichzeitig die Modelle ständig weiterzuentwickeln. Durch die technische Entwicklung konnten uns die Bausparkasssen im Laufe der Zeit auch detailliertere Daten zur Verfügung stellen.

Die Zielsetzung aller Modelle blieb aber im Grunde immer die gleiche, nämlich eine simulierte Fortschreibung der Zeitreihen des Bausparkollektivs, die unter Vorgabe von Steuerungsparametern sowohl eine Prognose der realen künftigen Weiterentwicklung erlaubt als auch das Durchspielen von verschiedenen Szenarien ermöglicht.

Im Folgenden sollen die Idee und die Umsetzung der wichtigsten in der Praxis eingesetzten Simulationsmodelle kurz erläutert werden, da wir einige der dort verwendeten Methoden im Zusammenhang mit Hidden-Markov-Modellen wieder aufgreifen werden. Verweise auf weitere ältere Modelle und kurze Beschreibungen dazu finden sich in [44].

3.2.1 Schichtenmodell

Im Schichtenmodell [15] wird das Bausparkollektiv durch wenige, von den Bausparkassen beobachtete typische Verhaltensmuster, den sogenannten Schichten, approximiert. Eine mit einem bestimmten Bausparsummenanteil versehene Schicht kann nach bauspartechnischen Regeln deterministisch durchgerechnet werden und liefert als Ergebnis Jahres-Zeitreihen der wichtigsten Bauspargrößen. Mittels eines nichtlinearen Optimierungsverfahrens werden die Anteile aller Schichten so bestimmt, dass jährlich erhobene Kollektivgrößen der Vergangenheit möglichst genau getroffen werden. Mit Hilfe der so gewonnenen Zusammensetzung der Schichten wird dann die Entwicklung des Bausparkollektivs in der Zukunft simuliert.

Das Schichtenmodell greift also nur auf wenige reale, kollektive Zeitreihen der Bausparkasse zurück – insbesondere werden keine Einzelvertragsdaten berücksichtigt – und liefert folglich nur eine grobe, nicht besonders realistische Abbildung der inneren Zusammensetzung des aktuellen Bestandes. Zudem ist die Realisierung nichtkonstanten Sparerverhaltens und flexibler Tarife mit diesem Modell recht schwierig. Es eignet sich somit weniger zur Prognose einer zukünftigen realen Entwicklung, sondern eher für qualitative Aussagen und zum Erkennen von generellen Wirkungsmechanismen bei Variation von Vorgaben und Parametern.

3.2.2 Mikrosimulationsmodell

Als Basis für das Mikrosimulationsmodell [24, 44] dienen sämtliche Einzelverträge einer Bausparkasse im Zeitverlauf. Im Gegensatz zum Schichtenmodell werden die Zeitreihen nicht deterministisch, sondern stochastisch extrapoliert, und die Daten der Vergangenheit werden nicht approximiert.

Die Grundannahme in diesem Modell besteht darin, dass die jährliche Weiterentwicklung eines Bausparvertrags als Markov-Kette aufgefasst werden kann; d. h. die Wahrscheinlichkeit, dass sich der Vertrag in einem Jahr in einem bestimmten Zustand befindet, hängt nur von den Größen des Vorjahres ab. Die möglichen Zustände des Markov-Prozesses entsprechen den verschiedenen Vertragszuständen, die ein Bausparvertrag annehmen kann. Zur Bestimmung der Übergangswahrscheinlichkeiten zwischen den Zuständen werden die Verträge anhand der Daten eines Referenzjahrgangs zunächst in Gruppen eingeteilt (s. u.). Das Verhalten der Verträge einer Gruppe im Jahr darauf führt zu Häufigkeitsverteilungen, aus denen dann die Übergangswahrscheinlichkeiten ermittelt werden. In der Simulation wird schließlich jedes einzelne Konto zufällig nach den Wahrscheinlichkeiten der Markov-Kette fortgeschrieben. Die Zuordnung der Verträge zu den Gruppen erfolgt dabei in jedem simulierten Jahr wieder neu, während die Wahrscheinlichkeitsverteilungen pro Gruppe konstant bleiben.

Nachdem in der ersten Version des Modells die Gruppierung anhand fest vorgegebener Merkmalsgrenzen erfolgt war, wurden in der Arbeit von Vannahme [44] erstmals verschiedene dynamische Clusterverfahren eingesetzt, wobei das *K*-means-Verfahren (siehe Abschnitt 3.2.3) für das Problem die besten Resultate lieferte. Als geeignete Clusterkriterien erwiesen sich hier Spargeldeingang, Anspargrad oder BWZ und die Bausparsumme des entsprechenden Referenzjahres.

Das Mikrosimulationsmodell eignet sich besonders für kurzfristige Prognosen, vor allem wegen der exakten Abbildung des Bestands zu Beginn der Simulation. Durch das Weiterrechnen jedes einzelnen Vertrags ist das Modell komplexer als das Schichtenmodell. Die oben beschriebene Neuzuordnung aller Verträge in jedem Simulationsjahr erschwert allerdings die geeignete Wahl der Modellparameter und eine Steuerung der Zeitreihen.

3.2.3 Mesoskopisches Modell

In dem aktuell von unserer Arbeitsgruppe eingesetzten mesoskopischen Modell [25] werden die mikroskopischen Daten und das Prinzip der makroskopischen Simulation aus den beiden oben

genannten Simulationsmodellen übernommen. Die Grundannahme des Modells besteht wieder darin, dass sich Verträge mit ähnlichem Muster im Sparverhalten für die Simulation zusammenfassen lassen. Neben einer Skizzierung des gesamten Modellverlaufs werden wir hier auf spezielle Aspekte der Modellierung, die auch für den Einsatz von Hidden-Markov-Modellen interessant sind, genauer eingehen.

Modellübersicht

Wie beim Schichtenmodell werden zunächst bestimmte Sparertypen definiert, die hier aber direkt aus den Daten der Einzelverträge mit vollständig bekannter Sparphase gewonnen werden. Dazu werden diese Verträge mittels der *K*-means-Methode, die auch schon im Mikrosimulationsmodell zur Clusterung eingesetzt wurde, zu Gruppen mit ähnlichem Sparverhalten im Zeitverlauf zusammengefasst. Für jedes dieser gewonnenen Cluster wird ein Prototyp errechnet, der die jeweilige Gruppe möglichst gut repräsentiert.

Jetzt können wir jeden Vertrag des Kollektivs, der sich zum Simulationsbeginn noch in der Sparphase befindet, demjenigen Prototyp zuordnen, der ihm gemessen an einem Abstandsmaß über dem Spargeldeingang am nächsten ist. Diese Zuordnung erfolgt mit Hilfe eines Netzwerkfluss-Algorithmus, bei dem die zu minimierenden Kosten auf den Kanten des Netzwerk-Graphen den Abständen der Verträge zu den Prototypen entsprechen. Da gerade bei sehr kurzen Verträgen die Abstände zu den verschiedenen Prototypen oft sehr nahe beieinander liegen, werden die Anteile der den Prototypen zugeordneten Verträge durch Kapazitätsschranken im Netzwerk gesteuert. Als Ergebnis der Zuordnung erhalten wir zu jedem Prototyp für jedes Abschlussjahr einen Anteil an allen Verträgen.

Durch statistische Untersuchungen auf den Daten werden schließlich das Zuteilungs- und das Tilgungsverhalten sowie ein mögliches Sonderverhalten der Prototypen bestimmt, so dass zusammen mit den Anteilen aus der Zuordnung wieder Schichten entstehen, die wie im Schichtenmodell deterministisch weitergerechnet werden können. Die oben nicht berücksichtigten Verträge, die ihre Sparphase schon beendet haben, werden in der Simulation in Sonderschichten abgewickelt, während ähnlich wie in beiden Vorgängermodellen in jedem simulierten Jahr nach vorgegebenen Anteilen auch wieder neue Schichten entstehen.

Clusterung mit dem *K*-means-Verfahren

Das *K*-means-Verfahren (auch Centroidmethode, Quadratfehlermethode; siehe z. B. [18, 44]) bestimmt bei gegebenem festem K eine Zerlegung $\mathcal{C} = (C_1, \ldots, C_K)$ von n Objekten $x_1, \ldots, x_n$ in K Cluster derart, dass die Summe der quadrierten Abstände der Objekte zu ihrem jeweiligen Mittelwertvektor z_k minimal wird. Die Objekte können dabei als Punkte im euklidischen Raum $\mathbb{R}^d$ aufgefasst werden, und als Abstandsfunktion wird i. d. R. die euklidische Metrik eingesetzt, so dass sich als zu minimierende Zielfunktion

$$f(\mathcal{C}) = \sum_{k=1}^{K} \sum_{x_i \in \mathcal{C}_k} \|x_i - z_k\|_2^2$$

ergibt. Für eine gegebene Clusterung wird die Zielfunktion minimal, wenn der Mittelwertvektor eines Clusters aus dem arithmetischen Mittel berechnet wird [44]:

$$z_k = \frac{1}{|C_k|} \sum_{x_i \in C_k} x_i \,, \qquad k = 1, \ldots, K \,.$$

Zur Lösung des Clusterproblems verwenden wir in erster Linie das Minimaldistanzverfahren, das folgendermaßen abläuft:

1. **Initialisierung:** Wahl einer Startpartition und Berechnung der Clustermittelpunkte
2. **Iteration:**
   (a) Zuordnung jedes Objekts zu seinem nächstgelegenen Clustermittelpunkt
   (b) Berechnung der neuen Clustermittelpunkte
3. **Abbruch**, sobald die Zielfunktion nicht mehr signifikant verbessert wird oder eine maximale Anzahl von Iterationen erreicht ist.

Das Verfahren konvergiert gegen ein (lokales) Minimum, da die Zielfunktion monoton fällt [44].

Zur Bestimmung von Prototypen für das Sparverhalten aus den Daten aller Verträge mit abgeschlossener Sparphase nehmen wir als Objekte zur Clusterung die jährlichen relativen Spargeldeingänge bezogen auf die Bausparsumme. Sei O_t^i der Spargeldeingang des Vertrags i im t-ten Vertragsjahr, dann wird das i-te Objekt zunächst durch den Vektor

$$x_i = (O_1^i, O_2^i, \ldots, O_{T_i}^i)$$

beschrieben, wenn der Vertrag T_i Jahre lang bespart wurde. Da der euklidische Abstand allerdings nur für Vektoren gleicher Dimension definiert ist, füllen wir die unterschiedlich langen Sequenzen bis $T = \max_{1 \leq i \leq n}(T_i)$ mit Nullen auf.

Die Clustermittelpunkte, die die Clusterung als Ergebnis liefert, entsprechen den mittleren jährlichen Spargeldeingängen eines Clusters und bilden fortan im Modell die Prototypen für das Sparverhalten des Kollektivs. Diese Prototypen ähneln den nach Erfahrungswerten der Bausparkassen erstellten Verhaltensmustern im Schichtenmodell (Regelsparer, Soforteinzahler, Niedrigsparer etc.), sind aber nach Konstruktion wesentlich besser den gegebenen Daten angepasst.

Modellbewertung

Durch das Aufsetzen des Simulationsmodells auf den realen Zeitreihen eines Kollektivs liefert das mesoskopische Modell bessere Ergebnisse als das Schichtenmodell, besonders im Bereich der mittelfristigen Prognosen. Gleichzeitig liegt die Komplexität des gesamten Modells weit niedriger als beim Mikrosimulationsmodell und ermöglicht deshalb das einfachere Durchrechnen von Szenarien bei Variation verschiedener Parameter.

Eine Schwäche des Modells liegt im starren, deterministischen Verhalten der Prototypen. So ergibt sich am Simulationsbeginn oft eine gewisse Unstetigkeit zwischen den realen Kollektivgrößen und den simulierten Zeitreihen, die sich durch das Zusammenfassen der zwar ähnlichen,

aber dennoch individuellen und mit einer bestimmten Varianz verteilten Daten der Einzelverträge zu festen Prototypen (Varianz = 0) erklären lässt. Daneben erschwert die Kombination verschiedener Komponenten – Clusterung nach Spargeldeingang, statistische Bestimmung verschiedener Häufigkeiten, Sonderbehandlung der Darlehensverträge etc. – die Handhabbarkeit des Modells.

Ansätze, neben dem Spargeldeingang noch weitere Merkmale in die Clusterung zu integrieren, wie z. B. Kündigung, Bausparsummenänderung oder Darlehensverzicht, haben gezeigt, dass es dann schwierig wird, ein geeignetes Abstandsmaß zu finden, mit dem diese Daten vergleichbar gemacht werden können. Diese schwer festzulegende Vergleichbarkeit unterschiedlicher Größen bereitet vor allem auch bei den variablen Längen der Merkmalsvektoren Probleme; die Längen der aus der Clusterung entstehenden Prototypenvektoren beeinflussen aber stark die Ergebnisse einer Simulation.

Alle diese Überlegungen führten schließlich zu der Idee, Hidden-Markov-Modelle zur Modellierung der Zeitreihen einzusetzen.

3.3 Einsatz von Hidden-Markov-Modellen

Hidden-Markov-Modelle werden seit Jahren erfolgreich zur Modellierung stochastischer Prozesse eingesetzt (vgl. Einleitung zu Kapitel 2). Gerade in der Spracherkennung konnten mit Hilfe dieser sehr flexiblen Modelle entscheidende Verbesserungen erzielt werden. So kann z. B. ein bestimmtes gesprochenes Wort mit Hilfe eines HMM modelliert werden, das einerseits die individuellen, vom jeweiligen Sprecher abhängigen Aussprachen abbildet, andererseits aber gerade die typischen, gemeinsamen Merkmale aller Aussprachen erkennt und modelliert. Mit einem so trainierten Modell kann mit einer hohen Genauigkeit erkannt werden, ob eine gesprochene Lautfolge das jeweilige Wort darstellen soll oder nicht. In der Praxis wird i. d. R. für jedes Wort, das erkannt werden soll, ein HMM trainiert, so dass letztendlich ein Klassifikator für die vorgegebene Menge von Wörtern vorliegt [38].

Ein weiteres Anwendungsgebiet für Hidden-Markov-Modelle sind die Analyse und die Modellierung von Zeitreihen [31]. Hier wird versucht, für bereits vorhandene Gruppen von Zeitreihen jeweils ein Modell zu finden, das die Daten am besten beschreibt, d. h. eine Klassifikation entfällt hier.

Betrachten wir die Bestimmung der Prototypen für das Sparverhalten im mesoskopischen Modell und die eigentliche Kollektivsimulation, sehen wir sofort Parallelen zu den beiden eben genannten HMM-Anwendungsgebieten: Wie in der Spracherkennung sollen aus den Daten der Spargeldeingänge typische Verhaltensmuster erkannt und klassifiziert werden, wobei innerhalb einer Klasse individuelle Abweichungen zu erwarten sind. Der Abhängigkeit einer Aussprache vom Sprecher, die über unbekannte interne Zustände abgebildet wird, steht hier die Abhängigkeit der Sparrate vom Sparer gegenüber. Andererseits besteht das Ziel einer Kollektivsimulation in der möglichst getreuen Abbildung von verschiedenen Zeitreihen mittels mathematischer Modelle, mit denen schließlich neue Zeitreihen erzeugt werden können. Der für uns wichtige letztere Aspekt der Datengenerierung, der in den beiden anderen Anwendungen keine Rolle spielt, spricht zusätzlich für den Einsatz von Hidden-Markov-Modellen.

3.3.1 Modellierungsidee

Der Grundgedanke besteht darin, die Prototypen des im vorigen Abschnitt vorgestellten mesoskopischen Modells durch einzelne Hidden-Markov-Modelle zu realisieren. Aus dieser Idee heraus könnte die Modellierung und Simulation von Zeitreihen eines Bausparkollektivs ungefähr folgendermaßen ablaufen:

Datenauswahl: Zuerst müssen die Daten bestimmt werden, die den Ausgabesymbolen der Hidden-Markov-Modelle entsprechen sollen. Dazu bieten sich alle Größen an, die in einem Vertrag nicht determiniert sind, wie z. B. der Spargeldeingang.

Sequenzen: Für jeden Vertrag, der in die Prototypen- bzw. Modellbildung einfließen soll, extrahieren wir aus den ausgewählten Daten eine Sequenz.

Modelle: Die Ausgabesymbole der Hidden-Markov-Modelle sind durch die Sequenzen festgelegt; die Modelltopologie dagegen – Anzahl von Zuständen, mögliche Übergänge, Start- und Endzustände etc. – ist nicht natürlich vorgegeben, sondern muss in geeigneter Weise und abhängig von der jeweiligen Anwendung ermittelt werden.

Training und Clusterung: Bei gegebener Einteilung der Sequenzen zu verschiedenen Gruppen kann nach dem Baum-Welch-Algorithmus jede Gruppe von Sequenzen zum Training eines HMM benutzt werden, das damit optimal den Daten angepasst wird (vgl. Abschnitte 2.2.3 und 2.5). Zur Bestimmung der Gruppen sollte ein Verfahren eingesetzt werden, das eine im Sinne der Modellierung „optimale“ Partition aller Sequenzen erstellt.

Simulation Mit den trainierten Hidden-Markov-Modellen können nach den entsprechenden Start-, Übergangs- und Ausgabeverteilungen neue Sequenzen erzeugt bzw. unvollständige verlängert werden, die damit eine Approximation für zukünftige (Teil-)Sequenzen darstellen. Je nachdem, welche Daten in die Sequenzen eingeflossen sind, liegen die gewünschten Zeitreihen direkt vor oder können durch bauspartechnische Berechnungen aus den generierten Sequenzen ermittelt werden.

Eine solche Modellierung greift auch Ansätze des Mikrosimulationsmodells auf, bei dem das Verhalten der Bausparer ebenfalls über einen Markov-Prozess abgebildet wurde.

In den nächsten Abschnitten werden die einzelnen Schritte des hier skizzierten Vorgehens detaillierter beschrieben. Dabei wollen wir uns aber direkt auf bestimmte Punkte beschränken, die in dieser Arbeit vorrangig behandelt werden sollen.

3.3.2 Auswahl der Daten und Sequenzen

Die Ausgaben eines HMM stellen das Ergebnis eines stochastischen Prozesses dar. Es liegt folglich nahe, als Daten der Modellierung diejenigen zu verwenden, die den variablen Aktionen des Bausparers entsprechen und in ihrer Gesamtheit als stochastische Größen interpretiert werden können. Die wichtigsten davon sind:

- relativer Spargeldeingang (SPE) pro Zeitraum
- Kündigung

- Bausparsummenänderung
- Fortsetzung (vorläufiger Verzicht auf die Zuteilung)
- Höhe und Zeitpunkt der Guthabensauszahlung
- Darlehensverzicht
- Höhe und Zeitpunkt des Darlehens
- Tilgungen pro Zeitraum

Mit Hilfe dieser Größen lässt sich ein Bausparvertrag zusammen mit den Tarif- und Kassenvorgaben (Zinssätze, Zuteilungsbedingungen etc.) durchrechnen, und daraus ergeben sich andere wichtige Informationen, wie zum Beispiel der Zeitpunkt der Zuteilung. Unter Verwendung aller aufgelisteten Daten ist somit ein Modell denkbar, das einen kompletten Bausparvertrag mit allen resultierenden Zeitreihen abbildet.

Ziel dieser Arbeit ist es jedoch, den prinzipiellen Einsatz von Hidden-Markov-Modellen an einem einfachen Modell zu untersuchen und deshalb die Komplexität möglichst gering zu halten. Da der Spargeldeingang den Verlauf eines Vertrags stark beeinflusst und schon in den in Abschnitt 3.2 vorgestellten Kollektivmodellen eine zentrale Rolle gespielt hat, werden wir als Ausgaben bzw. Sequenzen die jährlichen Spargeldeingänge in Prozent der Bausparsumme (SPE) betrachten:

$$\begin{aligned} O_t^i &:= \text{SPE des Sparers } i \text{ im Vertragsjahr } t, \text{mit } O_t^i \in [0, 100]\,, \\ T_i &:= \text{Anzahl der Sparjahre des Sparers } i\,, \\ O^i &= (O_1^i, O_2^i, \ldots, O_{T_i}^i)\,. \end{aligned}$$

Einer Sequenz, deren zugrunde liegender Vertrag die Sparphase durch Zuteilung bereits verlassen hat, hängen wir ein spezielles Endsymbol Θ an:

$$O^i = (O_1^i, O_2^i, \ldots, O_{T_i}^i, O_{T_i+1}^i := \Theta)\,.$$

In einer solchen „vollständigen“ SPE-Sequenz steckt gleichzeitig eine weitere Information: Die Länge der Sequenz bestimmt die Spardauer bzw. den relativen Zeitpunkt der Zuteilung in Vertragsjahren. Aus jeder gegebenen (Teil-)Sequenz $O_{(t)}^i = (O_1^i, O_2^i, \ldots, O_t^i)$ lassen sich zusammen mit den Vorgaben des jeweiligen Tarifs auch der Anspargrad und die BWZ des Vertrags am Ende jedes Jahres ableiten.

Da vollständige Sequenzen stets die Sparphase eines zugeteilten Vertrags wiedergeben, müssen die entsprechenden Verträge bei ihrer Zuteilung die tariflichen Mindestbedingungen erfüllt haben. Folglich liegt die Summe aller Einträge einer solchen Sequenz nie deutlich unter dem Mindestanspargrad (der Anspargrad berechnet sich aus den Spargeldern plus den Zinsen seit Vertragsbeginn, siehe Abschnitt 3.1). Auf diese wichtige Nebenbedingung aus den Daten kommen wir später noch zurück (vgl. Abschnitt 3.4).

Zum Trainieren eines HMM werden wir nur vollständige Sequenzen mit Endsymbolen einsetzen, da das resultierende Modell die komplette Sparphase abbilden soll.

3.3.3 Modellarchitektur

Unter dem Begriff Modellarchitektur verstehen wir die folgenden Elemente eines HMM:

- Anzahl der Zustände, N
- belegte Zustandsübergänge und Startwahrscheinlichkeiten ($\Rightarrow$ Start- oder Endzustände)
- Art der Verteilungen auf den Zuständen (diskret oder stetige Dichtefunktionen)
- Anzahl der Ausgabesymbole bzw. Anzahl der Mischverteilungen, M

Die beiden ersten Punkte – Anzahl von Zuständen, belegten Übergängen und Startwahrscheinlichkeiten – bezeichnen wir mit der Topologie oder der Struktur eines Modells (vgl. Abschnitt 2.1). Wir werden im Weiteren beide Begriffe synonym verwenden.

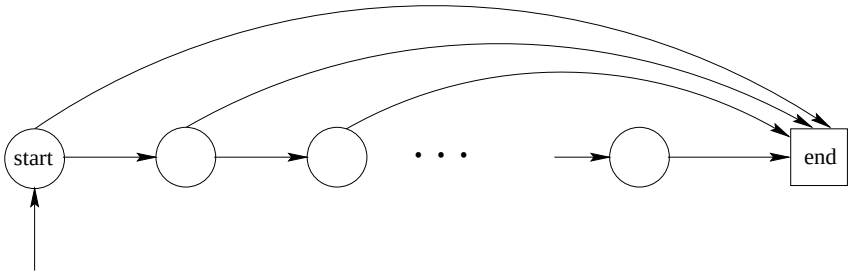
Nichtbelegte Zustandsübergänge oder Startwahrscheinlichkeiten werden durch Null-Einträge an entsprechender Stelle in A bzw. π spezifiziert (vgl. Abschnitt 2.2.3, letzter Absatz). Ein Startzustand S_i liegt dann vor, wenn $\pi_i > 0$ gilt; ein ausgezeichneter Endzustand S_j ist dadurch charakterisiert, dass für alle $k \in \{1, \ldots, N\}$ gilt: $a_{jk} = 0$. Bei Verwendung einer diskreten Dichtefunktion für die Ausgaben werden die M Ausgabesymbole immer auf die Zahlen 1 bis M abgebildet. Da wir die Daten auf die eindimensionale Größe SPE einschränken wollen, können im stetigen Fall nur univariate Dichtefunktionen eingesetzt werden. In der vorliegenden Arbeit werden wir außerdem bei allen Modellen einen ausgezeichneten Endzustand verwenden, der nur das Endsymbol Θ ausgeben kann.

Als Links-Rechts-Modelle werden alle Modelle bezeichnet, bei denen kein Wechsel in einen Vorgängerzustand möglich ist. Dabei legen wir eine feste Ordnung der Zustände von $1 \ldots N$ zugrunde. Falls ein solches Modell keine Selbstübergänge hat, nennen wir es zusätzlich strikt. In Abbildung 3.1(a) ist ein striktes Links-Rechts-Modell zu sehen, das für unsere Anwendung in dem Sinne minimal ist, als dass es das kleinste Modell dieser Art ist, das alle Sparsequenzen der Länge $T_{max} = \max_i(T_i)$ abbilden kann ($N = T_{max}+1$). In diesem Modell startet jeder Zustandspfad im Startzustand S_1 und kann nur aus direkten Nachfolgern und dem Endzustand bestehen. Somit entspricht der Zustand S_j genau dem j-ten Vertragsjahr.

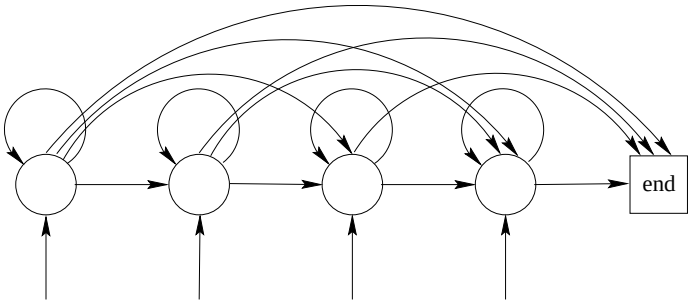
Im Modell (b) in Abbildung 3.1 sind dagegen die Zustände von den Vertragsjahren losgelöst. Es gibt vom Modell zudem keine Einschränkung der Sequenzlänge, da in dem Modellgraphen durch die Selbstübergänge Kreise enthalten sind (ein Kreis in einem Graphen ist eine alternierende Folge von Knoten und Kanten, bei der Start- und Zielknoten gleich sind). Mit solchen Modellen können deshalb auch Sequenzen modelliert werden, deren Länge größer ist als die Anzahl der Zustände im HMM ($N \leq T_{max}$).

Neben den Links-Rechts-Modellen werden wir in unseren Anwendungen auch vollständig verbundene Modelle untersuchen (vgl. Abschnitt 7.2). Grundsätzlich erhöht aber jeder zusätzliche freie Parameter den Rechenaufwand und verringert bei fester Anzahl von Trainingssequenzen die statistische Genauigkeit beim Schätzen der Parameter.

Da die Spargeldeingänge in den Sequenzen als Prozentzahlen beliebige Werte im Intervall $[0, 100]$ annehmen können, werden wir als Verteilungsfunktionen auf den Zuständen stetige univariate Normalverteilungen bzw. linksseitig gestutzte Normalverteilungen, wie sie in Kapitel 5 vorgestellt werden, einsetzen (siehe dazu auch Abschnitt 3.4).



(a) „Minimales“ striktes Links-Rechts-Modell



(b) Links-Rechts-Modell mit maximal vielen Übergängen

Abbildung 3.1: Beispiele zur Modelltopologie

Bei Einsatz von Hidden-Markov-Modellen mit diskreten Ausgaben muss das SPE-Intervall in M disjunkte Intervalle aufgeteilt werden und die ${O_t}^i$ müssen auf den zugehörigen Intervallindex transformiert werden. Bei großem M führt dies jedoch zu dünn besetzten Intervallen bei vielen zu trainierenden Parametern, während uns bei kleinem M die Ungenauigkeit bei Training und Simulation zu groß erscheint. Ein weiteres Argument gegen diskrete Ausgaben ist die Tatsache, dass bei einem HMM mit diskreten Ausgaben die Wahrscheinlichkeit für jedes Ausgabesymbol unabhängig von den anderen trainiert wird (vgl. Formeln in Anhang A). Somit geht bei der Intervallbildung die Nachbarschaftsinformation zweier Ausgabewerte, die sich nur gering unterscheiden, aber in verschiedenen diskreten Klassen landen, in den Modellen verloren. Bei stetigen Ausgaben dagegen sind die Ausgabewahrscheinlichkeiten immer über die Verteilungsfunktion verknüpft.

3.3.4 Training und Clusterung

Generell wäre es denkbar, für alle Sparsequenzen eines Kollektivs ein einziges großes HMM aufzustellen und zu trainieren. Sinnvoller dagegen ist eine Abbildung der verschiedenen Sparertypen mittels mehrerer Modelle. Dies hat vor allem den Vorteil, dass einzelne Gruppen analysiert und in einer Simulation besser gesteuert oder separat betrachtet werden können.

Da eine natürliche Einteilung in bestimmte Sparertypen nicht existiert, müssen wir wie schon beim mesoskopischen Modell eine Clusterung der Sequenzen durchführen (hier unterscheidet

sich unser Vorgehen vom Training von bestimmten Wörtern bei der Spracherkennung, denn dort ist eine Klassifizierung der Trainingssequenzen vorgegeben). Ein hierzu eingesetztes Clusterverfahren sollte die Modellierung der Sequenzen durch trainierte Modelle berücksichtigen. Es liegt nahe, analog zum Reestimierungsverfahren, bei dem die Likelihood $\prod_i L(O^i|\lambda)$ aller Sequenzen O^i eines Modells λ maximiert wird (vgl. Abschnitt 2.2.3), die Zielfunktion für das Clusterproblem ebenfalls in Abhängigkeit der $L(O^i|\lambda)$ zu formulieren. In Kapitel 4 werden wir ein HMM-basiertes Clusterverfahren vorstellen, das das Trainieren von Modellen unter Verwendung des Baum-Welch-Algorithmus und eine Clusterung der Sequenzen in geeigneter Weise kombiniert.

3.3.5 Simulation

Die trainierten Hidden-Markov-Modelle sollten bei geeigneter Wahl der zum Training eingesetzten Daten das statistische Verhalten des Bausparkollektivs repräsentieren. Sequenzen, die von solchen Modellen erzeugt werden, können folglich in ihrer Gesamtheit als Approximation einer zukünftigen Kollektiventwicklung dienen.

Das Erzeugen einer Sequenz erfolgt mittels Zufallszahlengenerierung nach den Verteilungsparametern des Modells:

1. **Initialisierung:** Bestimmen des Startzustands q_1 nach der diskreten Verteilung π
2. **Iteration:**
   (a) Erzeugen der Ausgabe des Zustands q_t nach der diskreten Verteilung b_{q_t} bzw. der stetigen Dichte f_{q_t}
   (b) Bestimmen des Zustands q_{t+1} nach den Einträgen der Übergangsmatrix A
3. **Abbruch**, sobald ein Endzustand erreicht ist oder die erzeugte Sequenz eine vorgegebene Länge T hat

In einer Kollektivsimulation der Zukunft müssen vor allem auch Verträge behandelt werden, die noch nicht abgewickelt sind und deshalb nur unvollständige Sequenzen liefern. Eine solche Sequenz O^i ordnen wir demjenigen HMM zu, das sie mit der größten Wahrscheinlichkeit reproduziert, denn dieses Modell wird erwartungsgemäß die beste Approximation für die weitere Entwicklung des stochastischen Prozesses und der daraus resultierenden Sequenz liefern. Wir bestimmen also das Modell, das von allen K Modellen die größte Likelihood $L(O^i|\lambda_k)$ aufweist.

Die unvollständige Sequenz kann jetzt genauso wie oben nach den Wahrscheinlichkeitsverteilungen des Modells vervollständigt werden. Dazu müssen wir den Zustand q_t festlegen, in dem der stochastische Prozess starten soll, wenn die Sequenz bereits aus t Ausgaben besteht. Diesen Zustand können wir z. B. mit

$$q_t := \operatorname*{argmax}_{1 \le i \le N} \alpha_t(i)$$

bestimmen, da $\alpha_t(i)$ die Wahrscheinlichkeit angibt, dass sich der Prozess nach Ausgabe der Teilsequenz $O_{(t)}$ im Zustand i befindet (vgl. Abschnitt 2.2.1).

Eine andere, deterministische Möglichkeit der Sequenzerzeugung besteht darin, beim Wechsel zwischen den Zuständen ähnlich wie beim Viterbi-Pfad nur in den jeweils wahrscheinlichsten zu springen und dort die Ausgabe mit der größten Wahrscheinlichkeit oder auch den Mittelwert als Folgesymbol zu nehmen. Eine so generierte Sequenz kann ähnlich wie im mesoskopischen Modell als Prototyp für das jeweilige Modell stehen. Der gegensätzliche Ansatz dazu ist, statt einzelne (Teil-)Sequenzen zu generieren auf algorithmischem Wege eine komplette Verteilungsfunktion für die möglichen Ausgabesequenzen eines Modells zu erstellen. Diese Verteilungsfunktion kann dazu benutzt werden, verschiedene statistische Wahrscheinlichkeitsaussagen zu treffen (Konfidenzintervalle, Kenngrößen etc.).

In der vorliegenden Arbeit werden wir die letztgenannten Ansätze nicht weiter verfolgen, da wir einerseits gerade das deterministische Element des mesoskopischen Modells verändern wollten und andererseits durch Generieren von ausreichend vielen zufälligen Sequenzen die entsprechende Verteilung approximieren können.

3.4 Erweiterungen für Hidden-Markov-Modelle

Bevor die in den vorigen Abschnitten beschriebenen Algorithmen und Methoden auf Originaldaten der Bausparkassen angewendet und anhand der Ergebnisse diskutiert werden sollen, behandeln wir in den nächsten Kapiteln zunächst die teilweise bereits angesprochenen HMM-Erweiterungen aus theoretischer Sicht, d. h. losgelöst von unserer konkreten Anwendung auf Zeitreihen eines Bausparkollektivs:

In Kapitel 4 wird ein HMM-basiertes Clusterverfahren formuliert, das nach bestimmten Optimalitätsbedingungen eine Menge von Sequenzen in K Gruppen einteilt und gleichzeitig K zugehörige Hidden-Markov-Modelle als Repräsentanten der Gruppen liefert. Anschließend schlagen wir einige Methoden zur Bewertung einer solchen Clusterung vor.

In Kapitel 5 untersuchen wir die Einsatzmöglichkeiten einer linksseitig, um den negativen Bereich gestutzten Normalverteilung als Dichtefunktion der Ausgaben. Dies hat den Hintergrund, dass wir als Sequenzen bei unseren Anwendungen die ausschließlich positiven Spargeldeingänge der Verträge einsetzen möchten. Bei einer Modellierung mit der in x-Richtung unbeschränkten Normalverteilung werden jedoch bei der Simulation, d. h. beim Generieren von Sequenzen, immer auch negative Ausgaben entstehen, die bauspartechnisch unsinnig sind.

In Kapitel 6 schließlich entwickeln wir eine neue Klasse von erweiterten Hidden-Markov-Modellen, die die deterministische Nebenbedingung der Zuteilung, der die vollständigen Spargeld-Sequenzen der Trainingsmenge unterliegen, besser abbilden können.

Die in den drei Kapiteln vorgestellten Erweiterungen werden wir in Kapitel 7 zur Umsetzung der in diesem Kapitel vorgeschlagenen Modellierungsidee zur Abbildung der Spargeld-Sequenzen einsetzen.

Kapitel 4

Clustern mit Hidden-Markov-Modellen

Methoden der Clusteranalyse werden in den unterschiedlichsten Disziplinen der Wissenschaft eingesetzt. Das grundsätzliche Ziel dieser Methoden besteht in der Gruppierung von Datenobjekten nach Ähnlichkeits- oder Unähnlichkeitskriterien, um somit die Objekte zu klassifizieren und gegebenenfalls Strukturen in den Daten zu erkennen. Eine Einteilung der Daten in verschiedene Gruppen oder Cluster kann auch als konzeptionelle Hilfe für den Umgang mit großen Datenmengen dienen. Von einer Clusterung nach Kriterien einer bestimmten Datenmodellierung erwarten wir eine bessere Modellbildung und somit eine höhere Qualität von Analyse und Simulation.

Der Einsatz von Clusterverfahren in den Simulationsmodellen für Bausparkollektive hat sich sowohl beim Mikrosimulationsmodell als auch beim mesoskopischen Modell bewährt (vgl. Abschnitt 3.2). Das Prinzip des in beiden Modellen verwendeten *K*-means-Verfahrens wird hier wieder aufgegriffen, wobei auch starke Parallelen zum Clustern mit Hilfe von Mischverteilungen vorliegen. Eine umfangreiche Zusammenstellung der verschiedenen Methoden und Anwendungen der Clusteranalyse findet sich z. B. in [18] und [11].

In den folgenden Abschnitten werden wir ein HMM-basiertes Clusterverfahren vorstellen, das auf der Maximierung der Likelihood aller Sequenzen bezüglich ihrer beschreibenden Hidden-Markov-Modelle beruht. Anschließend diskutieren wir verschiedene Aspekte des Verfahrens. Dazu gehört neben der Frage der Initialisierung und Wahl der Startmodelle vor allem das Problem der optimalen Clusteranzahl und der Bewertung bzw. Güte einer gefundenen Clusterung.

4.1 Problembeschreibung und Zielfunktion

Unser Ziel ist es, eine Menge von n Sequenzen möglichst gut mit K Hidden-Markov-Modellen abzubilden, d. h. wir suchen gleichzeitig eine Partitionierung der Sequenzen und die Parameter für die Modelle, die die verschiedenen Datencluster optimal beschreiben. Die Architektur und die Anzahl der Modelle sollen dabei vorgegeben und fest sein.

Für $K = 1$ reduziert sich das Problem auf das Training der Parameter und kann sofort mit Hilfe des Baum-Welch-Algorithmus gelöst werden; das resultierende HMM maximiert dann (lokal)

das Produkt der Likelihood $L(O^i|\lambda)$ über alle n Sequenzen. Es liegt deshalb nahe, bei $K > 1$ eine Gruppierung der Daten zu suchen, die ebenfalls das Produkt der $L(O^i|\lambda_k)$ maximiert, wobei für jede Sequenz die Likelihood des jeweiligen Cluster-Modells betrachtet wird. Wir formulieren das folgende Problem:

HMM-Cluster-Problem *Gesucht werden eine Partitionierung $\mathcal{C} = (C_1, C_2, \ldots, C_K)$ der Menge der n Sequenzen $\mathcal{O} := \{O^1, O^2, \ldots, O^n\}$ mit $C_1 \cup C_2 \cup \ldots \cup C_K = \mathcal{O}$ und Parameter $\lambda_1, \ldots, \lambda_K$ von korrespondierenden Hidden-Markov-Modellen, so dass die Zielfunktion*

$$f(\mathcal{C}) = \prod_{k=1}^{K} \prod_{O^i \in C_k} L(O^i|\lambda_k) \tag{4.1}$$

maximiert wird. Dabei sei K eine fest vorgegebene Anzahl von gesuchten Clustern.

4.2 Algorithmus

Das HMM-Cluster-Problem weist eine große Ähnlichkeit mit dem Clusteransatz mit Mischverteilungen auf [11,32], bei dem ebenso eine gesuchte Klasseneinteilung mit einer Parameterschätzung von vorgegebenen Datenmodellen verknüpft ist. Dort wird angenommen, dass jeder der zu klassifizierenden Vektoren $x_1, \ldots, x_n$ einer Komponentenverteilung $f_k(x|\theta_k)$ mit unbekannten Parametern θ_k entstammt. Der Datenraum wird dadurch durch eine Mischverteilung mit der Dichte

$$f(x) = \sum_{k=1}^{K} c_k f_k(x|\theta_k)$$

beschrieben, wobei die Gewichte c_k die unbekannte Wahrscheinlichkeit beschreiben, dass ein Vektor x der k-ten Mischkomponente entstammt. Dieses Clusterproblem wird mit einem iterativen Verfahren gelöst, das abwechselnd jeden Vektor der Komponente k zuordnet, der er mit der größten Wahrscheinlichkeit angehört, $k = \text{argmax}_j[c_j f_j(x|\theta_j)]$, und anschließend die unbekannten Parameter θ_k getrennt für jede Mischkomponente durch Maximum-Likelihood-Schätzung bestimmt.

Das in Abschnitt 3.2.3 beschriebene Minimaldistanzverfahren zur Lösung des K-means-Problems zeigt die gleiche Struktur: Zuordnung jedes Objektes zu dem Cluster, dessen Mittelwertvektor am nächsten liegt, und Minimierung der Abstände aller Objekte eines Clusters zu ihrem Mittelwertvektor durch dessen Neuberechnung. In der Tat wird in [17] gezeigt, dass das K-means-Verfahren als Spezialfall des Clusterverfahrens mit Mischungen von Normalverteilungen aufgefasst werden kann.

In Anlehnung an diese beiden Verfahren formulieren wir einen Algorithmus zur Lösung des HMM-Cluster-Problems.

4.2.1 Maximum-Likelihood-Verfahren

Gegeben seien die Sequenzen $O^1, \ldots, O^n$, eine feste Clusterzahl K und die Architektur von K Hidden-Markov-Modellen. Dann bezeichnen wir den folgenden Algorithmus als Maximum-Likelihood-Verfahren:

1. **Initialisierung ($t = 0$):** Wahl der Startparametersätze $\lambda_1^0, \ldots, \lambda_K^0$ für die K Modelle.
2. **Iteration ($t \in \{1, 2, \ldots\}$):**
   (a) Erzeugung einer neuen Partition der Sequenzen, indem jede Sequenz O_i zu dem Modell zugeordnet wird, das die Likelihood $L(O_i|\lambda_k^{t-1})$ maximiert
   (b) Bestimmung der neuen Parameter $\lambda_1^t, \ldots, \lambda_K^t$ durch Reestimieren der Modelle mit den jeweils zugeordneten Sequenzen und den Startparametern $\lambda_1^{t-1}, \ldots, \lambda_K^{t-1}$ nach dem Baum-Welch-Algorithmus
3. **Abbruch**, sobald die Zielfunktion (4.1) nicht mehr signifikant verbessert wird (ε-Schranke), keine Sequenz mehr das Modell wechselt oder eine vorgegebene maximale Anzahl I_c von Iterationen erreicht ist

Lemma 4.1 *Die Zielfunktion* (4.1) *des Maximum-Likelihood-Verfahrens wächst monoton.*

Beweis: Seien $\mathcal{C}^t$ die nach der Iteration t gefundene Partitionierung, λ_k^t die entsprechenden trainierten Modellparameter und $\log f(\mathcal{C})$ die logarithmierte Zielfunktion aus (4.1). Dann gilt

$$\begin{aligned}
\log f(\mathcal{C}^t) &= \sum_{k=1}^{K} \sum_{O^i \in C_k^t} \log L(O^i|\lambda_k^t) \\
&\leq \sum_{k=1}^{K} \sum_{O^i \in C_k^t} \max_{l=1\ldots K} \log L(O^i|\lambda_l^t) \\
&= \sum_{k=1}^{K} \sum_{O^i \in C_k^{t+1}} \log L(O^i|\lambda_k^t) \\
&\leq \sum_{k=1}^{K} \sum_{O^i \in C_k^{t+1}} \log L(O^i|\lambda_k^{t+1}) \\
&= \log f(\mathcal{C}^{t+1}),
\end{aligned}$$

und somit ist auch $f(\mathcal{C}^t) \leq f(\mathcal{C}^{t+1})$.

Die letzte Ungleichung folgt aus der Tatsache, dass für die Likelihood eines Modells mit beliebigem Initial-Parameter λ und trainiertem Parameter $\bar{\lambda}$ nach dem Baum-Welch-Algorithmus gilt: $L(O|\bar{\lambda}) \geq L(O|\lambda)$ (vgl. Abschnitte 2.2.4, 2.3 und 2.5); die λ_k^{t+1} entsprechen aber gerade den mit den Startwerten λ_k^t trainierten Parametern.

□

Lemma 4.2 *Die Ausgabedichten $b_i(x)$ der Hidden-Markov-Modelle des Maximin-Likelihood-Verfahrens seien nach oben beschränkt. Dann ist die Zielfunktion* (4.1) *ebenfalls nach oben beschränkt.*

Beweis: $f(\mathcal{C})$ ist ein endliches Produkt von Likelihood-Funktionen, die sich wiederum als endliche Summe von Produkten aus den Übergangswahrscheinlichkeiten und Dichtefunktionen zusammensetzen:

$$\begin{aligned} L(O|\lambda) &= \sum_{s\in S} \pi_{s_1} b_{s_1}(O_1) \prod_{\tau=2}^{T} a_{s_{\tau-1}s_\tau} b_{s_\tau}(O_\tau) \\ &\leq \sum_{s\in S} \prod_{\tau=1}^{T} b_{s_\tau}(O_\tau)\,, \end{aligned}$$

da die π_i und a_{ij} als Wahrscheinlichkeiten immer ≤ 1 sind. Nach Voraussetzung sind alle $b_{s_\tau}(O_\tau)$ nach oben beschränkt und somit gilt dies auch für $L(O|\lambda)$.

□

Aus Lemma 4.1 und Lemma 4.2 folgt sofort:

Satz 4.3 *Das Maximum-Likelihood-Verfahren konvergiert unter den Annahmen aus Lemma 4.2 (Beschränktheit der HMM-Ausgabedichten nach oben) auch für $I_c = \infty$.*

Der Zielfunktionswert muss jedoch selbst bei unendlicher Laufzeit nicht das globale Maximum erreichen, da der Algorithmus sich von einem lokalen Maximum nicht wegbewegen kann. Durch die ebenfalls nur lokale Maximierung der Modellparameter im Baum-Welch-Algorithmus und deren Abhängigkeit von den Initialwerten können wir desweiteren nicht ausschließen, dass eine feste, aber beliebige Partition von Sequenzen im Laufe des Verfahrens mehrmals entsteht. Die maximale Iterationszahl I_c schließlich garantiert aber, dass das Verfahren nach endlicher Zeit stoppt.

4.2.2 Laufzeit

Zur Abschätzung der Laufzeit unterscheiden wir zwischen diskreten und stetigen Ausgaben bei den Modellen. Zur Vereinfachung gehen wir davon aus, dass die Anzahl der Zustände N und die Anzahl der Symbole bzw. Mischkomponenten M für alle Modelle gleich sind. Mit $T := \max_i(T_i)$ bezeichnen wir die maximale Sequenzlänge.

Diskrete Ausgaben:

In einem Schritt des Reestimierungsalgorithmus werden zunächst für jede Sequenz alle Forward- und Backward-Variablen in

$$O(T_i N^2)$$

Schritten und danach die Parameter π, A und B gemeinsam für alle n_k Sequenzen des k-ten Modells in

$$O(n_k TN \max[N, M])$$

Schritten berechnet (vgl. Anhang A). Die Laufzeit für die Reestimierung aller Cluster in Schritt 2(b) beträgt folglich mit $n = n_1 + \ldots + n_K$:

$$O(I_r\, n\, T\, N\ \max[N, M])\,.$$

I_r stellt dabei eine Konstante für die maximale Iterationsanzahl beim Reestimierungsalgorithmus dar, da wir die Abhängigkeit dieser Anzahl von den Eingabedaten und Modellgrößen nicht kennen.

Die Zuordnung aller Sequenzen zu dem Modell mit der jeweils höchsten Likelihood in Schritt 2(a) geschieht in

$$O(n\, K\, T\, N^2)$$

Schritten, da für jede Sequenz die Likelihood zu jedem Modell berechnet werden muss. Die Gesamtlaufzeit des Maximum-Likelihood-Verfahrens beträgt somit

$$O(I_c\, n\, T\, N(N\, K + I_r \max[N, M]))\,,$$

wobei I_c wieder die Iterationskonstante des Verfahrens bezeichnet.

Stetige Ausgaben:

Bei Verwendung von Modellen mit stetigen Ausgaben erhöht sich die Komplexität zur Bestimmung der Forward-Backward-Variablen auf $O(T\, N^2 M)$, während zur Berechnung der Reestimierungsparameter eines Modells $O(n_k\, T\, N^2 M)$ Operationen durchgeführt werden müssen. Die Gesamtlaufzeit erhöht sich deshalb auf

$$O(I_c\, n\, T\, N^2 M(K + I_r))\,.$$

4.2.3 Wahl der Startmodelle

Da das Maximum-Likelihood-Verfahren meist in einem lokalen Maximum endet, spielt die Wahl der Startparameter der verwendeten Hidden-Markov-Modelle keine unwesentliche Rolle. Daneben muss vor der Anwendung des Clusterverfahrens auch die Architektur der einzelnen Modelle festgelegt werden. Diese sollte sich nach der Struktur und Zusammensetzung der Daten richten bzw. muss im konkreten Fall getestet werden (vgl. Abschnitt 3.3.3 und Kapitel 7).

Für die Belegung der Startparameter schlagen wir folgende Möglichkeiten vor:

- zufällige Belegung aller Parameter unter Einhaltung der stochastischen Nebenbedingungen für die diskreten Wahrscheinlichkeiten (vgl. Abschnitt 2.1)

- gleichförmige Belegung der Übergangs- und Startwahrscheinlichkeiten; Belegung der Ausgabeparameter durch zufällige Variation der statistischen Mittelwerte und Varianzen aus den Daten (z. B. pro Zeitraum t)
- identische Belegung der Parameter aller Modelle und anschließendes Training mit vorgruppierten Sequenzen (Gruppierung zufällig oder durch ein anderes Clusterverfahren gewonnen, z. B. nach dem K-means-Verfahren)

Grundsätzlich muss darauf geachtet werden, dass im ersten Iterationsschritt des Verfahrens jede Sequenz von mindestens einem Modell erzeugt werden kann, denn eine Sequenz, bei der dies nicht der Fall ist, kann auch nicht beim Training der Modelle berücksichtigt werden. Aus diesem Grund sollten bei den Zufallsinitialisierungen neben den stochastischen Nebenbedingungen auch untere Schranken für die Parameter eingehalten werden, so dass die Initialmodelle nicht zu sehr spezialisiert sind.

In Kapitel 7 werden wir für unsere Anwendungen verschiedene Initialisierungen der Modellparameter testen und vergleichen.

4.2.4 Modifikationen

Bei der Anwendung des Maximum-Likelihood-Verfahrens kann es vorkommen, dass einem HMM in Schritt 2(a) keine Sequenzen zugeordnet werden. Dieses HMM wird dadurch in Schritt 2(b) nicht trainiert und erhält folglich auch im weiteren Verlauf des Verfahrens keine Sequenzen. Um dies zu vermeiden ordnen wir einem solchen Modell diejenigen zwei Sequenzen zu, die von allen Sequenzen die schlechtesten Likelihoodwerte $L(O|\lambda)$ zu ihrem jeweiligen Modell λ aufweisen. Dabei achten wir darauf, dass die Cluster, aus denen wir die Sequenzen wegnehmen, danach nicht leer sind.

Ein solches Vorgehen verschlechtert zunächst die Zielfunktion (4.1); da sich jedoch das HMM mit den neu zugeordneten Sequenzen beim anschließenden Training stark auf diese spezialisieren kann, ist die Zielfunktion danach i. d. R. größer als vor dem Austausch der Sequenzen, und dies war auch in allen unseren Anwendungen der Fall. Allerdings kann durch die nur lokale Konvergenz des Baum-Welch-Algorithmus die Konvergenz des in dieser Form modifizierten Maximum-Likelihood-Verfahrens nicht mehr garantiert werden.

In der Praxis führt das Verschieben von nur *einer* Sequenz i. d. R. zu einer so starken Spezialisierung des HMM, dass danach keine andere Sequenz mehr zu diesem Modell zugeordnet wird, während bei dem oben beschriebenen Vorgehen die Anzahl der zu diesem HMM gehörigen Sequenzen im Laufe der folgenden Iterationen wächst.

Eine weitere Modifikation des Algorithmus besteht darin, im Schritt 2(b) die Hidden-Markov-Modelle in einem ersten Durchgang nicht bis zur (lokalen) Konvergenz zu trainieren, sondern jeweils nach wenigen Baum-Welch-Schritten abzubrechen. Nach dem Erfüllen des Abbruchkriteriums (Schritt 3) starten wir das Verfahren bei diesmal vollständiger Reestimierung ein zweites Mal. Da die Likelihood nach *jedem* Trainingsschritt im Baum-Welch-Algorithmus größer oder gleich der vorherigen ist (vgl. Abschnitt 2.2.4), bleibt für diese Modifikation des Verfahrens die Konvergenzaussage aus Satz 4.3 erhalten.

Das vorzeitige Abbrechen des Modelltrainings verbessert in der Praxis trotz der gestiegenen Anzahl der Iterationsschritte 2(a) und 2(b) die Geschwindigkeit des Verfahrens, da im Baum-Welch-Algorithmus die größten Parameteränderungen in den ersten Schritten erfolgen und beim zweiten Durchlauf des Verfahrens meist nur noch wenige Sequenzen umsortiert werden.

4.3 Bewertung von Clusterungen

Die Bewertung von Clusterverfahren bzw. von konkreten Ergebnissen der jeweiligen Verfahren ist ein schwieriges Thema, da es im Allgemeinen keine objektiven Kriterien für die Güte einer Clusterung gibt. Während eine Clusterung von wenigen, niedrig dimensionalen Daten eventuell noch optisch begutachtet und qualitativ eingeschätzt werden kann, sind wir bei großen Datenmengen auf mathematische Bewertungsverfahren angewiesen, die uns zumindest einen Anhaltspunkt für die Qualität einer vorliegenden Clusterung geben können.

In der Literatur finden sich zahlreiche verschiedene Ansätze zur Bewertung von Clusterungen und Clusterverfahren, die von statistischen Tests auf das Vorliegen einer signifikanten Struktur in den Daten bis zu speziellen Indizes für ganz bestimmte Aspekte einer Clusterung reichen. In [18] werden zahlreiche dieser Methoden ausführlich beschrieben.

Wir werden im Folgenden verschiedene Kriterien für das HMM-Clusterverfahren diskutieren und in Anlehnung an bekannte Bewertungsverfahren eigene Kenngrößen entwickeln, wobei wir im Wesentlichen nur relative Maßzahlen betrachten, die keine absoluten Qualitätsaussagen liefern, sondern immer nur Clusterungen im Verhältnis zu anderen Clusterungen bewerten. Damit soll uns vor allem die Entscheidung erleichtert werden, wie wir die folgenden unbekannten Größen bei der Anwendung des Maximum-Likelihood-Verfahrens setzen müssen:

- Clusteranzahl K
- Anzahl der Zustände und Übergänge (Topologie)
- Initialparameter

Die in den nächsten Abschnitten vorgestellten Maße werden wir jeweils beschreiben und mit Hilfe von verschiedenen Clusterungen eines künstlich generierten Testdatensatzes (Abbildung 4.1) erläutern. Die 500 Testdaten entsprechen Sequenzen der Länge 2 und sind offensichtlich stark strukturiert, so dass wir hier eine relativ klare Vorstellung von einer optimalen Clusteranzahl $K = 5$ haben. In der Tat sind die Daten mit 5 speziellen Modellen vom Typ 1 in Abbildung 4.2 erzeugt worden. Zur Clusterung der Testdaten werden wir jedoch beide Modelle aus Abbildung 4.2 mit normalverteilten Ausgaben (nur eine Mischkomponente) verwenden. Die Initialisierungsparameter wählen wir für alle Cluster-Modelle gleich und starten das Maximum-Likelihood-Verfahren mit einer zufälligen Gruppierung der Testdaten.

An dieser Stelle wird bereits eine Schwierigkeit bei der Verwendung von Hidden-Markov-Modellen deutlich: Modell 2 ist von der Topologie her wesentlich vielseitiger als Modell 1, und es ist nicht klar, ob die Testdaten nicht mit weniger Modellen des Typs 2 repräsentiert werden können.

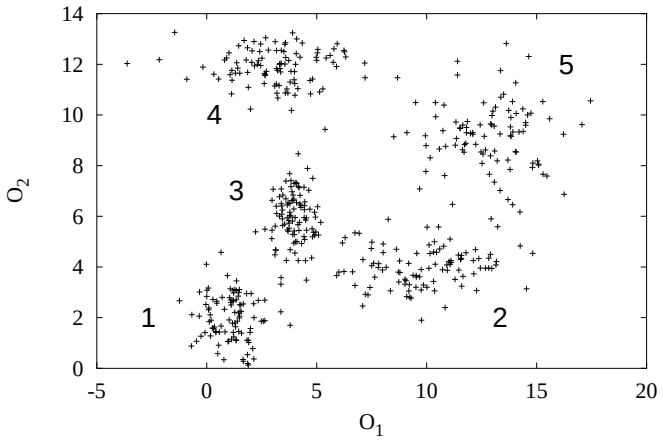


Abbildung 4.1: Geometrische Interpretation der strukturierten Testdaten mit je zwei Einträgen (O_1, O_2)

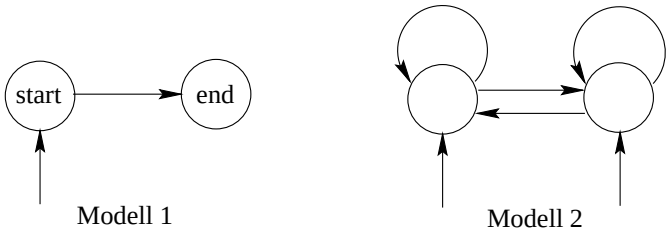


Abbildung 4.2: Modellstrukturen zum Clustern der Testpunkte

Da ein HMM eine probabilistische Modellierung darstellt, kann eine Clusterung mit dem Maximum-Likelihood-Verfahren im Allgemeinen nicht nach geometrischen Maßstäben beurteilt werden.

Die nachfolgenden Auswertungen von Clusterungen stellen nur Beispiele mit sehr speziellen Testsequenzen dar und greifen nur einige Aspekte einer Clusterung auf; wir werden die Praxistauglichkeit der entwickelten Kenngrößen ausführlicher in Kapitel 7 anhand der Zeitreihen von Sparzahlungen in einem Bausparkollektiv diskutieren.

4.3.1 Zielfunktion

Die Zielfunktion (4.1) des Maximum-Likelihood-Verfahrens bzw. die Summe der logarithmierten Likelihoodwerte am Ende einer Clusterung,

$$L_{\sum} = \sum_{k=1}^{K} \sum_{O^i \in C_k} \log L(O^i | \lambda_k) \,, \tag{4.2}$$

kann als relative Kenngröße für die Übereinstimmung der Daten mit den zugehörigen Modellen betrachtet werden. Abbildung 4.3 zeigt den typischen wachsenden Verlauf dieser Größe bei Va-

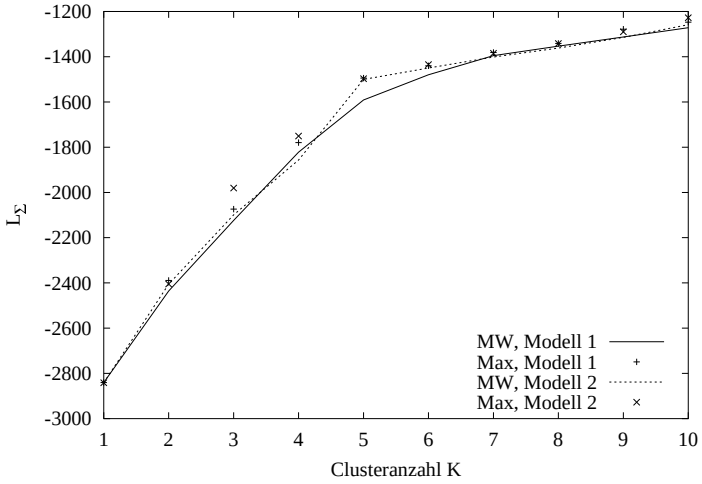


Abbildung 4.3: Mittlere und maximale $L_\sum$ bei zehn Läufen pro K

riation der Clusteranzahl. Die Testdatenstruktur von fünf natürlichen Clustern wird bei Verwendung von Modell 1 nicht signifikant angezeigt. Ab $K = 5$ ist bei beiden Kurven ein schwächeres Ansteigen der Likelihood als vorher zu verzeichnen; eine Präferenz für eine bestimmte Clusteranzahl kann jedoch nicht unbedingt harausgelesen werden.

4.3.2 Abstandsmaße zwischen Hidden-Markov-Modellen

Ein Aspekt einer vorliegenden HMM-Clusterung ist die Ähnlichkeit der trainierten Cluster-Modelle bzw. die statistische Trennung der K Cluster. Zwei Modelle mit großer statistischer Übereinstimmung sprechen eher für eine Clusterung mit weniger Modellen bzw. für eine Zusammenlegung dieser beiden Modelle.

Als Maß für die Ähnlichkeit zweier Hidden-Markov-Modelle wird in [22] folgendes probabilistisches Abstandsmaß vorgestellt:

$$D(\lambda_1, \lambda_2) := \lim_{T\to\infty} \frac{1}{T} \left[\log L(O^1_{(T)}|\lambda_1) - \log L(O^1_{(T)}|\lambda_2)\right] ,$$

wobei die Sequenz $O^1_{(T)}$ der Länge T vom HMM λ_1 erzeugt wurde. D vergleicht damit, wie gut die Modelle λ_1 und λ_2 eine Sequenz beschreiben, die vom Modell λ_1 erzeugt wurde. Aus dem unsymmetrischen D lässt sich mittels

$$D_s(\lambda_1, \lambda_2) := \frac{1}{2} [D(\lambda_1, \lambda_2) + D(\lambda_2, \lambda_1)]$$

das symmetrische Abstandsmaß D_s gewinnen. Die Konvergenzrate der Distanzfunktion D ist abhängig von der Modellarchitektur; in der Praxis muss T deshalb ausreichend groß gewählt werden. Bei Modellen, die keine beliebig langen Sequenzen erzeugen können, müssen entsprechend viele Sequenzen aneinander gehängt werden.

HMM	*1*	*2*	*3*	*4*	*5*	*6*
1	0.0	16.8	8.4	48.8	40.0	32.6
2		0.0	13.4	37.9	8.5	29.7
3			0.0	16.8	28.8	13.6
4				0.0	12.7	0.8
5					0.0	13.2

Tabelle 4.1: Abstände D_s der Modelle einer Clusterung (Typ 1, $K = 6$)

Tabelle 4.1 zeigt die berechneten Werte für D_s nach einer Clusterung der Testdaten mit 6 Modellen vom Typ 1 (vgl. Abbildungen 4.1 und 4.2). Dabei wurden für jede Abstandsberechnung wieder neue Sequenzen mit einer Gesamtlänge $T = 5000$ zufällig erzeugt. Auffällig ist der sehr kleine Abstand von 0.8 zwischen dem HMM 4 und dem HMM 6. Tatsächlich bilden beide Modelle Sequenzen der Datenwolke 4 in der linken oberen Ecke in Abbildung 4.1 ab und ähneln sich stark in den trainierten Modellparametern. Die zu den Modellen 1, 2, 3 und 5 zusortierten Sequenzen entsprechen recht genau den Datenwolken mit der jeweils gleichen Nummer. Wir stellen fest, dass sich in diesem Beispiel die berechneten Abstände geometrisch interpretieren lassen; der Abstand zwischen HMM 1 und HMM 5 ist z. B. relativ groß, und auch die Datenwolken 1 und 5 liegen weit auseinander.

Die D_s-Abstände der Modelle untereinander sollen zusätzlich dazu verwendet werden, um in Anlehnung an einen bei geometrischen Clusterverfahren eingesetzten Index eine weitere relative Kenngröße zum Vergleich verschiedener Clusterungen aufzustellen. Der Davies-Bouldin-Index [18] setzt bei einer geometrischen Clusterung die Kompaktheit der Cluster in Beziehung zur Separierbarkeit der Cluster. Zunächst werden zwei Cluster C_j und C_k verglichen:

$$R_{jk} := \frac{e(C_j) + e(C_k)}{d(C_j, C_k)} . \tag{4.3}$$

Darin ist $e(C_j)$ der durchschnittliche Abstand der Daten aus Cluster C_j zu ihrem Clustermittelpunkt und $d(C_j, C_k)$ der Abstand der Mittelpunkte von C_j und C_k. Der Index nach Davies-Bouldin berechnet sich dann aus

$$DB := \frac{1}{K} \sum_{k=1}^{K} \max_{j \neq k} (R_{jk}) . \tag{4.4}$$

Kleine Werte für DB sprechen für eine kompaktere Clusterung, denn dann ist der Durchmesser der Cluster im Verhältnis zum Abstand der Cluster untereinander klein (siehe [18, 44]).

In unserem Fall möchten wir die bei einer HMM-Clusterung entstandenen Modelle hinsichtlich ihrer statistischen Ähnlichkeit bewerten. Wir konstruieren deshalb folgenden **DB-Index**:

$$DB_{hmm} := \frac{1}{K} \sum_{k=1}^{K} \max_{j \neq k} \left(\frac{1}{D_s(\lambda_j, \lambda_k)} \right) . \tag{4.5}$$

So wie ein großes R_{jk} aus (4.3) auf eine starke Überlappung der beiden Cluster C_j und C_k hindeutet, entspricht ein großer Wert für $1/D_s(\lambda_j, \lambda_k)$ einer größen Ähnlichkeit der Modelle λ_j und λ_k und somit einer starken Übereinstimmung der Sequenzen, die beide Modelle beschreiben bzw. erzeugen können. Der in dieser Form konstruierte DB-Index berücksichtigt aber im Gegensatz zum Davies-Bouldin-Index (4.4) nur die trainierten Modelle einer Clusterung und nicht die Sequenzen.

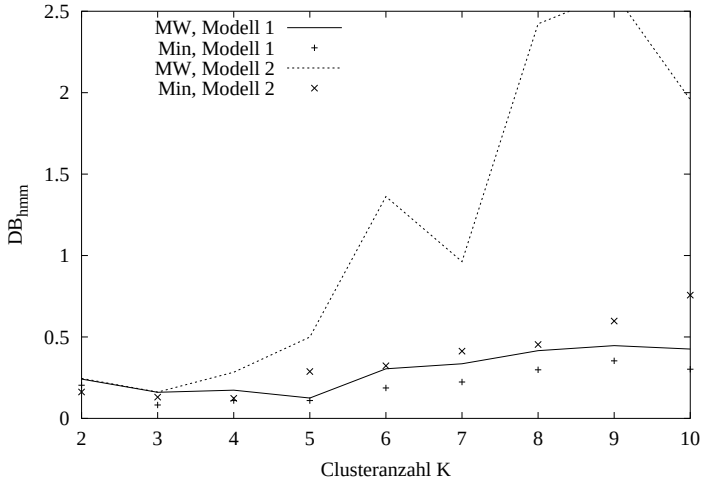


Abbildung 4.4: Mittlere und minimale DB-Indizes bei 10 Läufen pro K

In Abbildung 4.4 ist der DB-Index unter Verwendung der Testdaten von Seite 48 und der Modelle von Seite 48 für wachsende Clusteranzahl K aufgetragen. Die Kurve von Modell 1 weist für $K = 5$ ein Minimum auf, während eine Clusterung mit Modellen vom Typ 2 zu weniger Clustern tendiert. Die Punkte der minimalen *DB*-Werte zeigen allerdings auch, dass der Index stark von der jeweiligen Clusterung bzw. Initialisierung abhängt.

Eine andere, in [30] beschriebene Klasse von Abstands- und Ähnlichkeitsmaßen, die alternativ zum Abstandsmaß D_s für die Beurteilung einer Clusterung eingesetzt werden kann, basiert auf der Wahrscheinlichkeit, dass zwei Hidden-Markov-Modelle mit diskreten Ausgaben unabhängig voneinander die gleiche Sequenz ausgeben (co-emission probability):

$$A(\lambda_1, \lambda_2) := \sum_{O \in \Sigma^*} P(O|\lambda_1) P(O|\lambda_2) \,.$$

Dabei bezeichnet Σ^* den Raum aller möglicher Sequenzen, die mit dem Ausgabealphabet Σ gebildet werden können. Für Links-Rechts-Modelle kann $A(\lambda_1, \lambda_2)$ mit Hilfe eines Algorithmus der dynamischen Programmierung in $O(N_1 N_2)$ Schritten berechnet werden, wobei N_i die Anzahl der Zustände in Modell i ist [30]. Da der Algorithmus jedoch nicht ohne weiteres auf Modelle mit stetigen Ausgaben übertragen werden kann, wurde diese Abstandsklasse nicht weiter betrachtet.

4.3.3 Datenverteilung auf den Zuständen bei stetigen Ausgaben

Von einer guten HMM-Clusterung sollte man erwarten, dass die berechneten Dichtefunktionen der Ausgaben auf den Zuständen der trainierten Modelle möglichst gut die Daten repräsentieren, die zum Training der Ausgabeparameter der Zustände beigetragen haben. Deshalb liegt es nahe, bei einer gegebenen Clusterung die Verteilung dieser Daten mit der zugehörigen trainierten Verteilungs- bzw. Dichtefunktion zu vergleichen.

Die Datenverteilung eines Modellzustands s wird aus allen Ausgaben O_t^i der Sequenzen eines Modells k gebildet, die mit einer Wahrscheinlichkeit größer null von diesem Zustand ausgegeben werden (bei Ausgabe der gesamten Sequenz O^i) und somit zum Training der Ausgabedichte verwendet wurden. Diese Wahrscheinlichkeit einer Ausgabe O_t^i kann mit

$$g_t^i(s) := \alpha_t^i(s)\,\beta_t^i(s) \tag{4.6}$$

leicht berechnet werden und entspricht der Gewichtung von Mittelwert und Varianz bei den Reestimierungsformeln (vgl. Abschnitt 2.2.1, Anhang A).

Für einen festen Zustand sei J die Anzahl der Ausgabedaten O_t^i mit $g_t^i(s) > 0$. Wir sortieren die O_t^i in aufsteigender Reihenfolge und normieren die Gewichte so, dass ihre Summe eins ergibt. Statt mit den Indizes i und j kennzeichnen wir die Ausgaben und ihre Gewichte jetzt nach ihrer Sortierung, und damit erhalten wir J Datenpaare (O_j, g_j), aus denen die empirische Verteilungsfunktion $F_J(x)$ gebildet werden kann: Unter der Annahme, dass die Ausgaben Realisationen einer Zufallsvariablen X sind, setzen wir $P(X \leq O_j) = \sum_{j'=1}^{j} g_{j'}$ und $P(X \leq x) = P(X \leq O_j)$ für $O_j \leq x < O_{j+1}$, und damit ergibt sich für die Verteilungsfunktion von X die Form der Kurve $F_J(x)$ aus Abbildung 4.5.

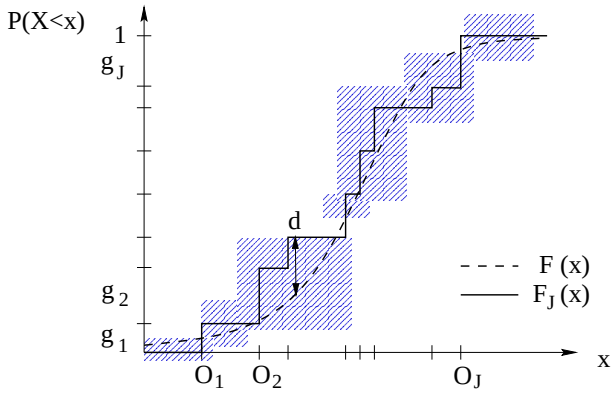


Abbildung 4.5: Empirische und analytische Datenverteilung eines Zustands

KS-Index

Ein Standardverfahren der Statistik zum Testen der Hypothese, ob eine Datenreihe einer bestimmten Verteilungsfunktion entstammt, ist der Kolmogorov-Smirnov-Test (K-S-Test, [35] und [37], Kap. 14.3). Bei diesem Test wird als Testgröße die maximale absolute Differenz zwischen

der empirischen Verteilungsfunktion $F_J(x)$ der J Daten und der analytischen Verteilungsfunktion $F(x)$ betrachtet (vgl. Abbildung 4.5):

$$D = \max_{-\infty < x < \infty} \left| F_J(x) - F(x) \right| .$$

Unter der Annahme, dass beide Verteilungen übereinstimmen (Hypothese H_0), kann die Verteilung von D selbst approximativ berechnet werden, d. h. wir kennen die Wahrscheinlichkeit, dass die Zufallsgröße D größer ist als der beobachtete Wert d: $\hat{P} = P(D > d)$. Bei kleinem Signifikanzwert $\hat{P}$ wird man die Hypothese H_0 verwerfen.

Größere Signifikanzwerte $\hat{P}$ sprechen eher dafür, dass die empirische und die analytische Verteilung identisch sind. Wir definieren daher ein Maß für die Güte einer Clusterung als Funktion der Signifikanzwerte $\hat{P}_s^k$ aus den K-S-Statistiken aller Zustände $s = 1, \ldots, N^k$ der an der Clusterung beteiligten Modelle $k = 1, \ldots, K$, den sogenannten **KS-Index**:

$$KS := \sum_{k=1}^{K} \sum_{s=1}^{N^k} \omega_s^k \hat{P}_s^k . \tag{4.7}$$

Als Wahrscheinlichkeiten liegen die $\hat{P}_s^k$ alle zwischen 0 und 1. Die ω_s^k dienen einer zusätzlichen Gewichtung der Zustände, die von den Sequenzen mit unterschiedlichen Wahrscheinlichkeiten frequentiert werden. Durch Summation der Gewichte $g_t^i(s)$ aus (4.6) erhalten wir zunächst eine Größe, die äquivalent ist zu der Wahrscheinlichkeit, dass ein Zustand s die Ausgabesymbole der Sequenzen des Clusters k ausgibt:

$$w_s^k := \sum_{O_i \in C_k} \sum_{t=1}^{T_i} g_t^i(s) \tag{4.8}$$

Damit setzen wir

$$\omega_s^k := \frac{w_s^k}{\sum_{k'=1}^{K} \sum_{s'=1}^{N^k} w_{s'}^{k'}} . \tag{4.9}$$

Der KS-Index ist somit bezüglich der Anzahl der Zustände und der Modelle skaliert und nimmt Werte zwischen 0 und 1 an. Bei dem Vergleich zweier Clusterungen entspricht der größere Wert der besseren Clusterung bezüglich der modellierten Datenverteilungen auf den Zuständen.

CS-Index

Ein weiteres Verfahren zum Testen der Hypothese, ob eine Datenreihe einer bestimmten Verteilungsfunktion entstammt, ist der χ^2-(Anpassungs-)Test ([35] und [37], Kap. 14.3). In diesem Fall werden diskrete oder diskretisierte Verteilungen betrachtet, deren r Ausprägungen wir als Klassen bezeichnen. p_i sei die Wahrscheinlichkeit der analytischen Verteilung für eine Klasse i. Die Testgröße wird aus den quadrierten Differenzen zwischen den tatsächlich beobachteten Daten einer Klasse (H_i) und den nach der Verteilung erwarteten Daten der Klasse (Jp_i, wenn J Daten vorliegen) gebildet:

$$T = \sum_{i=1}^{r} \frac{(H_i - Jp_i)^2}{Jp_i} .$$

Wieder unter der Annahme, dass beide Verteilungen übereinstimmen (Hypothese H_0), ist T näherungsweise χ^2-verteilt (mit $r-1$ Freiheitsgraden), und damit kann der Signifikanzwert $\bar{P} = P(T > t)$ für einen beobachteten Wert t berechnet werden. Ein kleiner Wert $\bar{P}$ spricht gegen die Hypothese H_0.

Zur Verwendung des χ^2-Tests diskretisieren wir die empirische und die analytische Verteilung für die Ausgaben der Zustände und definieren dann analog zum KS-Index den **CS-Index** aus den entsprechenden Signifikanzwerten aller Zustände in allen Modellen:

$$CS := \sum_{k=1}^{K} \sum_{s=1}^{N^k} \omega_s^k \bar{P}_s^k \,. \tag{4.10}$$

Die Gewichte ω_s^k setzen wir wieder nach Gleichung (4.9), und damit gelten für den CS-Index die gleichen Aussagen wie für den KS-Index.

In Abbildung 4.6 sind der KS- und der CS-Index bei Variation der Clusteranzahl aufgetragen, wobei wieder die Testdaten und Modelle von Seite 48 und 48 zugrunde liegen. Die natürliche Clusterstruktur wird weitgehend durch ein Maximum von den Indizes angezeigt; anders als beim DB-Index werden hier aber eher Clusterungen mit mehr Clustern positiv bewertet (vgl. Abbildung 4.4).

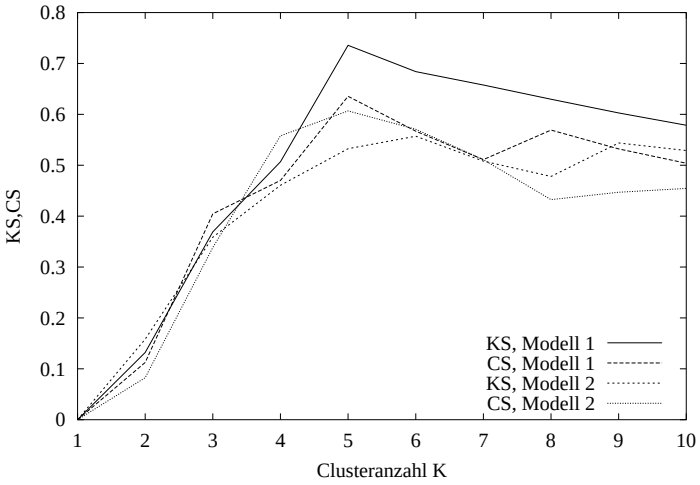


Abbildung 4.6: Mittlere KS- und CS-Indizes bei zehn Läufen pro K

ID-Index

Die Verwendung der statistischen Tests hat den Nachteil, dass man bei komplexen Daten und wenigen Clustern meist sehr kleine, zum Vergleich nicht sehr aussagekräftige Werte für den KS-Index und den CS-Index bekommt. Zudem ist der CS-Index abhängig von der Diskretisierung der Verteilungen, und vom KS-Index wird eine bessere Anpassung der Verteilungen nicht registriert, solange sich der maximale Abstand der beiden Verteilungsfunktionen nicht ändert. Dies ist z. B. der Fall, wenn sich viele Daten um einen Wert konzentrieren.

Wir betrachten deshalb als Alternative die Integraldifferenz der beiden Verteilungsfunktionen für jeden Zustand s und jedes Modell k (vgl. Abbildung 4.5, schraffierte Fläche):

$$ID_s^k = \int_{-\infty}^{\infty} |F_J^{(k,s)}(x) - F^{(k,s)}(x)|\, dx\,.$$

Mit diesen Größen und den Gewichten ω_s^k aus (4.9) definieren wir als dritte Variante den **ID-Index** (Integral-Differenz-Index) einer Clusterung:

$$ID := \sum_{k=1}^{K} \sum_{s=1}^{N^k} \omega_s^k ID_s^k\,. \tag{4.11}$$

Der ID-Index ist nach unten durch 0 beschränkt, und im Gegensatz zu den beiden anderen Indizes zeigt hier ein kleinerer Wert die bessere Clusterung an. Auch dieser Index liefert bei der Clusterung mit den Testdaten zufriedenstellende Ergebnisse (Abbildung 4.7), wobei der Extrempunkt der Kurven beider Modellen wieder bei $K = 5$ liegt.

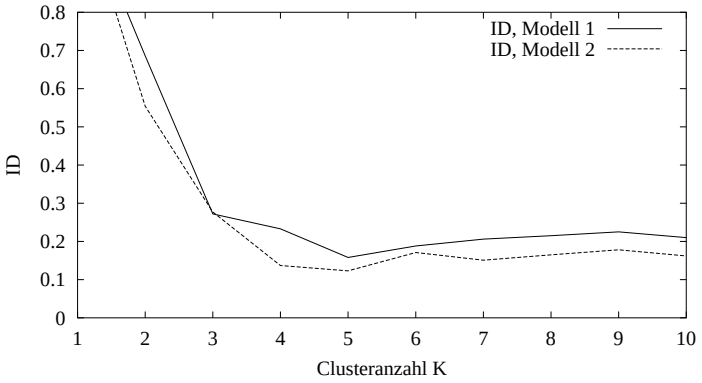


Abbildung 4.7: Mittlere ID-Indizes bei zehn Läufen pro K

Es sei noch darauf hingewiesen, dass bei der Berechnung des ID-Indexes eine gewünschte Tendenz zu weniger Clustern und Zuständen (vgl. Abschnitt 4.3.4) eingeht, die sich aus der Abhängigkeit der Größe ID_s^k von der Datenanzahl ergibt. Denn bei gleichbleibend gutem Zusammenpassen der Daten und der Verteilungsfunktion verbessert sich dieser Wert bei steigender Anzahl von Daten. Je weniger Zustände und Cluster aber eingesetzt werden, desto größer ist die Anzahl der Daten, die von einem Zustand ausgegeben werden.

4.3.4 Modellierung und Simulationsgüte

Im Zusammenhang mit der jeweiligen Anwendung und dem Einsatz einer HMM-Clusterung ergeben sich oft schon gewisse Bedingungen an die Clusterung. So ist uns in einem Simulationsmodell meist auch daran gelegen, mit wenigen Modellen und Parametern auszukommen, um

die Komplexität bei der Handhabung minimal zu halten. Daneben erfordern die Anzahl und die Struktur der Daten eine bestimmte Clustergröße, denn je weniger Daten zum Training eines Modells beigetragen haben, desto höher sind dessen Spezialisierung und die statistische Unsicherheit.

Die bisher betrachteten Kriterien zur Beurteilung einer HMM-Clusterung basieren alle auf den trainierten Modellen und den Daten, die geclustert wurden. Mit den Hidden-Markov-Modellen, die die Cluster beschreiben, können auch Sequenzen erzeugt werden, die wiederum mit den Trainingsdaten verglichen werden können. Dieses Vorgehen bietet sich gerade bei unserer Anwendung an, da wir die Clusterung nur als Hilfsmittel benutzen und nicht primär nach Strukturen in den Daten suchen, sondern hauptsächlich diese Daten modellieren und die Modelle zu Simulationszwecken einsetzen möchten. Der Ansatz besteht also darin, eine Clusterung nach der Eignung der entstandenen Modelle zur weiteren Verwendung zu bewerten. Eine solche Bewertung ist zwangsläufig abhängig von der konkreten Anwendung.

Eine Beurteilung der generierten Sequenzen hat zudem den Vorteil, dass auf diese Weise auch Modellierungsfehler zutage treten können. Ein HMM, das gegenüber einem anderen relativ gut trainiert wurde, kann trotzdem „falsche“ Sequenzen generieren, nämlich dann, wenn bestimmte Modellannahmen (Topologie, Verteilungsfunktion der Ausgaben etc.) getroffen wurden, durch die die Trainingsdaten nur unzureichend abgebildet werden können (vgl. Einleitung zu Kapitel 5 und Abschnitt 7.5).

Zum Vergleich von Trainingssequenzen und generierten Sequenzen können alle Verteilungen und statistischen Größen herangezogen werden, die sich sowohl aus den Sequenzen selbst als auch aus davon abgeleiteten Größen ergeben, wie z. B. die für eine Simulation eines Bausparkollektivs wichtigen Zeitreihen Anspargrad und BWZ (vgl. Abschnitt 3.1). Im Allgemeinen muss anhand der jeweiligen Anwendung entschieden werden, welche Größen und Verteilungen für die Modellierung und die Simulation, und somit für die Bewertung der Clusterung, wichtig sind.

Wir werden in Kapitel 7 im Rahmen der Clusterung von Spargeld-Sequenzen einer Bausparkasse verschiedene bauspartechnisch relevante Verteilungen und Kenngrößen vorstellen und diese jeweils für die Originaldaten und die trainierten Daten erstellen und vergleichen.

Kapitel 5

HMM mit gestutzten Normalverteilungen

Bei fast allen uns bekannten Anwendungen wird in Hidden-Markov-Modellen mit stetigen Ausgaben als Dichtefunktion auf den Zuständen die Normalverteilung bzw. eine Mischverteilung mehrerer Normalverteilungen eingesetzt. Die Zeitreihen der Spargeldeingänge, die in dieser Arbeit modelliert werden sollen, enthalten jedoch nur positive, reelle Zahlen. Wird ein HMM mit einfachen normalverteilten Ausgaben mit solchen positiven Daten trainiert, stellt sich typischerweise der Effekt ein, den wir in Abbildung 5.1 sehen: Die gewichteten relativen Häufigkeiten der

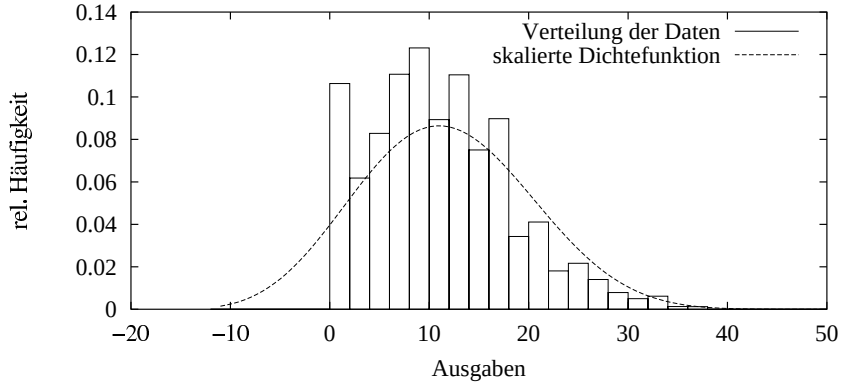


Abbildung 5.1: Datenverteilung und trainierte Dichte auf einem Zustand

Daten, die zum Training der Dichte-Parameter eines Zustands beigetragen haben (vgl. Abschnitt 4.3.3), überdecken nur einen Teil der korrespondierenden trainierten Dichtefunktion. Ein so trainierter Zustand stellt eine schlechte Datenmodellierung dar und wird insbesondere bei einer Simulation mit nicht zu vernachlässigender Wahrscheinlichkeit negative Ausgaben erzeugen. Das Beispiel verdeutlicht, dass wir zur Modellierung der Sparbeiträge statt der Normalverteilung, deren Dichte in ganz $\mathbb{R}$ strikt positiv ist, eine stetige Verteilung einsetzen sollten, die nur für positive x Werte größer 0 annehmen kann.

Die in der Literatur im Zusammenhang mit Hidden-Markov-Modellen theoretisch behandelten Verteilungsfamilien reichen von einfachen log-konkaven Dichtefunktionen [2, 3] über Verteilungen mit elliptisch-symmetrischen Dichten [29] bis hin zu multivariaten Mischverteilungen und Produkten von Mischverteilungen mit log-konkaven oder elliptisch-symmetrischen Dichten [20, 21]. Allerdings sind dort außer bei der Normalverteilung keine expliziten Reestimierungsformeln zur Bestimmung der entsprechenden Verteilungs-Parameter angegeben. Auch bei den bekannten Anwendungsgebieten und veröffentlichten Arbeiten fanden wir keine Hinweise auf in der Praxis eingesetzte stetige Verteilungen für positive Daten.

In den folgenden Abschnitten werden wir deshalb untersuchen, welche Verteilungsfunktionen mit der gewünschten Eigenschaft $P(X < 0) = 0$ bei Hidden-Markov-Modellen mit stetigen Ausgaben sinnvoll eingesetzt werden können. Dazu wird zunächst ein kurzer Abriss gegeben, wie sich bei stetigen Dichtefunktionen die Reestimierungsformeln für die Modellparameter ableiten lassen. Danach werden wir diese Formeln speziell für eine linksseitig bei $x = 0$ gestutzte Normalverteilung anpassen, wobei sich zeigen wird, dass diese sich trotz einiger analytisch und numerisch schwieriger Details in der Praxis erfolgreich einsetzen lässt. Im letzten Abschnitt werden wir kurz auf andere Dichtefunktionen eingehen und die Gründe nennen, warum wir diese nicht verwenden wollen.

Die in diesem Kapitel gewonnenen Erkenntnisse und hergeleiteten Verfahren bezüglich der gestutzten Normalverteilung werden wir in Kapitel 7 zur Modellierung der Spargeld-Zeitreihen einsetzen.

5.1 Allgemeine Herleitung der Reestimierungsformeln bei stetigen Ausgaben

Um zu verdeutlichen, an welcher Stelle wir ansetzen müssen, wenn wir das HMM mit einer speziellen Dichtefunktion versehen wollen, betrachten wir an dieser Stelle etwas genauer als in Abschnitt 2.3 die Herleitung der Formeln für die Parameter einer Mischverteilung von stetigen Dichtefunktionen, wie sie in [20] beschrieben ist. Diese Formeln gelten für das Trainieren mit einer einzelnen Sequenz, lassen sich aber leicht auf mehrere Sequenzen übertragen.

Wir beschränken uns im Weiteren auf univariate Funktionen, d. h. wir betrachten wie im diskreten Fall nur Modelle mit eindimensionalen Ausgaben auf den Zuständen. Dann hat die Mischverteilung im Zustand j die Form

$$b_j(x) = \sum_{m=1}^{M} c_{jm}\, b_{jm}(x; \mu_{jm}, \sigma_{jm}^2)\,, \tag{5.1}$$

mit $x, \mu_{jm} \in \mathbb{R}$, $c_{jm}, \sigma_{jm} \in \mathbb{R}^+$ und $\sum_{m=1}^{M} c_{jm} = 1$. Wir implizieren, dass die Mischkomponente b_{jm} durch die Parameter μ_{jm} und σ_{jm} eindeutig festgelegt ist. Ein HMM mit stetigen Ausgabeverteilungen nach (5.1) enthält neben dem Vektor der Startwahrscheinlichkeiten π und der Matrix der Übergangswahrscheinlichkeiten A als Parameter die $N \times M$-Matrizen $C = \{c_{jm}\}$, $\mu = \{\mu_{jm}\}$ und $U = \{\sigma_{jm}^2\}$. Als Kurzschreibweise für das dadurch bestimmte HMM verwenden wir wieder λ mit $\lambda = (\pi, A, C, \mu, U)$.

Es bezeichne $\Omega_s = \{1, \ldots, N\}$ die Indexmenge der Modellzustände und Ω_s^T deren T–tes kartesisches Produkt, so dass für eine Zustandsfolge $Q = (q_1 \cdots q_T)$ der Länge T gilt: $Q \in \Omega_s^T$, $q_t \in \Omega_s$. Analog dazu bezeichnen wir mit $\Omega_k = \{1, \ldots, M\}$ die Indexmenge der Mischkomponenten des Modells und mit Ω_k^T die Menge aller T-Tupel $K = (k_1 \cdots k_T)$ mit $k_t \in \Omega_k$. K nennen wir eine Mischkomponentenfolge.

Wir betrachten nun die Dichte bzw. Likelihood einer beobachteten Sequenz $O = (O_1 O_2 \cdots O_T)$ bei gegebenem λ (vgl. Abschnitt 2.3):

$$L(O|\lambda) = \sum_{Q \in \Omega_s^T} L(O, Q|\lambda) \,.$$

Für die gemeinsame Dichte einer Sequenz O und einer Zustandsfolge Q gilt mit der oben eingeführten Notation

$$\begin{aligned} L(O, Q|\lambda) &= \pi_{q_1} \left[\sum_{k=1}^{M} c_{q_1 k} b_{q_1 k}(O_1) \right] \prod_{t=2}^{T} \left[a_{q_{t-1} q_t} \sum_{k=1}^{M} c_{q_t k} b_{q_t k}(O_t) \right] \\ &= \sum_{K \in \Omega_k^T} \pi_{q_1} c_{q_1 k_1} b_{q_1 k_1}(O_1) \prod_{t=2}^{T} a_{q_{t-1} q_t} c_{q_t k_t} b_{q_t k_t}(O_t) \,. \end{aligned}$$

Der letzte Ausdruck zeigt, dass eine Zustandsfolge wiederum als Überlagerung von M^T Mischkomponentenfolgen interpretiert werden kann (diese Tatsache spiegelt sich auch wider bei der Generierung von Zufallszahlen einer Mischverteilung, vgl. letzter Absatz in Abschnitt 2.3). Wir können somit die gemeinsame Dichte einer Sequenz O, einer Zustandsfolge Q und einer Mischkomponentenfolge K definieren:

$$L(O, Q, K|\lambda) = \pi_{q_1} c_{q_1 k_1} b_{q_1 k_1}(O_1) \prod_{t=2}^{T} a_{q_{t-1} q_t} c_{q_t k_t} b_{q_t k_t}(O_t) \,, \tag{5.2}$$

und damit gilt

$$L(O|\lambda) = \sum_{Q \in \Omega_s^T} \sum_{K \in \Omega_k^T} L(O, Q, K|\lambda) \,.$$

Für zwei Modelle λ und λ' definieren wir die Hilfsfunktion $\mathcal{Q}(\lambda, \lambda')$:

$$\mathcal{Q}(\lambda, \lambda') := \sum_{Q \in \Omega_s^T} \sum_{K \in \Omega_k^T} L(O, Q, K|\lambda) \, \log L(O, Q, K|\lambda') \,. \tag{5.3}$$

Ähnlich wie beim Konvergenzbeweis des Baum-Welch-Algorithmus in Abschnitt 2.2.4 wird in [20] gezeigt, dass aus $\mathcal{Q}(\lambda, \lambda') \geq \mathcal{Q}(\lambda, \lambda)$ auch $L(O|\lambda') \geq L(O|\lambda)$ folgt.

Der Reestimierungs-Algorithmus startet nun mit einem Initialwert λ_0 für die Modellparameter und bestimmt in jedem Schritt ausgehend von den aktuellen Parametern λ neue Modellparameter $\bar{\lambda}$ mit der Eigenschaft, dass $\bar{\lambda}$ jeweils die Funktion $\mathcal{Q}(\lambda, \lambda')$ bezüglich λ' maximiert. Durch diese

iterative Transformation $\bar{\lambda} = \mathcal{T}(\lambda)$ wird solange die Likelihood $L(O|\lambda)$ verbessert, bis ein Fixpunkt erreicht ist, der zugleich einen kritischen Punkt der Likelihood darstellt und somit einem (lokalen) Maximum entspricht [20].

Die erforderliche Maximierung der Q-Funktion vereinfacht sich durch deren Zerlegbarkeit in mehrere Summanden, in denen die Modellparameter getrennt vorliegen. Denn nach (5.2) gilt

$$\log L(O, Q, K|\lambda') = \log \pi'_{q_1} + \sum_{t=1}^{T-1} \log a'_{q_t q_{t+1}} + \sum_{t=1}^{T} \log c'_{q_t k_t} + \sum_{t=1}^{T} \log b'_{q_t k_t}(O_t)\,,$$

und damit zerfällt die Q-Funktion folgendermaßen:

$$\begin{aligned}\mathcal{Q}(\lambda, \lambda') &= \sum_{Q\in\Omega_s^T} \sum_{K\in\Omega_k^T} L(O, Q, K|\lambda) \left[\log \pi'_{q_1} + \sum_{t=1}^{T-1} \log a'_{q_t q_{t+1}} + \sum_{t=1}^{T} \log c'_{q_t k_t} + \sum_{t=1}^{T} \log b'_{q_t k_t}(O_t)\right] \\ &=: \mathcal{Q}_\pi(\lambda, \pi') + \mathcal{Q}_a(\lambda, A') + \mathcal{Q}_c(\lambda, C') + \mathcal{Q}_b(\lambda, B')\,,\end{aligned}$$

wobei $B' = (\mu', U')$ die Parametermenge bezeichnet, über die alle Dichtefunktionen b'_{jm} definiert sind. Diese Teilfunktionen können jetzt individuell bezüglich ihrer Parameter π', A', C' bzw. B' maximiert werden. Die Funktionen $\mathcal{Q}_\pi$, $\mathcal{Q}_a$ und $\mathcal{Q}_c$ nehmen unter den stochastischen Nebenbedingungen $\sum_i \pi_i = \sum_j a_{ij} = \sum_m c_{jm} = 1$ ihr eindeutiges globales Maximum genau dann an, wenn π'_i, a'_{ij} und c'_{jm} nach den bekannten Gleichungen (2.9a) und (2.9b) bzw. (2.25c) gesetzt werden (vgl. Abschnitt 2.2.4). Der Summand $\mathcal{Q}_b$ erfordert eine eingehendere Diskussion. Zunächst gilt

$$\begin{aligned}\mathcal{Q}_b(\lambda, B') &= \sum_{Q\in\Omega_s^T} \sum_{K\in\Omega_k^T} L(O, Q, K|\lambda) \sum_{t=1}^{T} \log b'_{q_t k_t}(O_t) \\ &= \sum_{j=1}^{N} \sum_{m=1}^{M} \sum_{Q} \sum_{K} L(O, Q, K|\lambda) \sum_{t=1}^{T} \log b'_{q_t k_t}(O_t) \delta_{q_t j} \delta_{k_t m} \\ &= \sum_{j=1}^{N} \sum_{m=1}^{M} \sum_{t=1}^{T} L(O, q_t = j, k_t = m|\lambda) \log b'_{jm}(O_t)\,,\end{aligned}$$

wobei δ_{ij} das Kronecker-Symbol bezeichnet mit $\delta_{ij} = 1$ für $i = j$ und $\delta_{ij} = 0$ sonst. Damit ist $\mathcal{Q}_b(\lambda, B')$ mit $\vartheta_{jm} := (\mu_{jm}, \sigma^2_{jm})$ und $\zeta_t(j, m)$ aus (2.24c) wiederum eine Summe von unabhängigen Funktionen der Form

$$\begin{aligned}\mathcal{Q}_\vartheta(\lambda, \vartheta'_{jm}) &:= \sum_{t=1}^{T} L(O, q_t = j, k_t = m|\lambda) \log b'_{jm}(O_t; \vartheta'_{jm}) \\ &= L(O|\lambda) \sum_{t=1}^{T} \zeta_t(j, m) \log b'_{jm}(O_t; \vartheta'_{jm})\,. \qquad (5.4)\end{aligned}$$

Für in ϑ_{jm} streng log-konkave Funktionen $b_{jm}(x)$ mit $\lim_{|\vartheta_{jm}|\to\infty} \log b_{jm}(x) = -\infty$ ist leicht zu sehen, dass $\mathcal{Q}_\vartheta(\lambda, \vartheta'_{jm})$ ein eindeutiges Maximum besitzt (die Summe log-konkaver Funktionen ist

ebenfalls log-konkav). Für elliptisch-symmetrische Funktionen $b_{jm}(x)$ gilt das unter bestimmten, in der Praxis i. d. R. nicht einschränkenden Annahmen ebenso, wie in [20] gezeigt wird. Unter diese Kategorien fallen z. b. die Dichtefunktionen der Gamma-Verteilung und der Normalverteilung.

Die Lösung des Maximierungsproblems berechnet sich im Allgemeinen durch Differentiation, d. h. es werden Parameter ϑ'_{jm} bestimmt, für die gilt

$$\nabla_{\vartheta'_{jm}} \mathcal{Q}_{\vartheta}(\lambda, \vartheta'_{jm}) = \sum_{t=1}^{T} L(O, q_t = j, k_t = m|\lambda) \frac{\nabla_{\vartheta'_{jm}} b'_{jm}(O_t; \vartheta'_{jm})}{b'_{jm}(O_t; \vartheta'_{jm})} = 0\,. \tag{5.5}$$

Bei Verwendung einer speziellen Familie von Dichtefunktionen für das HMM muss also generell das globale Maximum der Funktion $\mathcal{Q}_{\vartheta}(\lambda, \vartheta'_{jm})$ bestimmt werden und daraus ergeben sich die entsprechenden Reestimierungsformeln. Für Dichtefunktionen, die weder log-konkav noch elliptisch-symmetrisch sind, muss dazu zunächst noch gezeigt werden, dass dieses globale Maximum existiert und eindeutig ist, denn nur dann ist die Konvergenz der iterativen Reestimierung gewährleistet. Die hier beschriebene Anpassung werden wir in den folgenden Abschnitten für gestutzte Normalverteilungen vornehmen.

5.2 Einsatz von gestutzten Normalverteilungen

Aus den stetigen Verteilungen, deren Dichtefunktionen auf der negativen x-Achse gleich 0 sind, greifen wir die linksseitig bei $x = 0$ gestutzte Normalverteilung heraus, da diese sich letztendlich als Ausgabeverteilung der Zustände eines HMM einsetzen lässt, wie wir in den folgenden Abschnitten sehen werden. Alternativen zu dieser Dichtefunktion werden wir in Abschnitt 5.3 diskutieren.

Da bei den in der vorliegenden Arbeit betrachteten Anwendungen ausschließlich eindimensionale Sequenzen modelliert werden sollen, beschränken wir uns im Weiteren auf Mischverteilungen von eindimensionalen (univariaten) gestutzten Normalverteilungen.

5.2.1 Notationen

Seien für $x, \mu \in \mathbb{R}$ und $\sigma > 0$

$$f(x; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}}\, e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

die Dichtefunktion einer (μ, σ^2)-verteilten Normalverteilung und

$$\bar{f}(x; \mu, \sigma) = \begin{cases} 0 & x < 0 \\ \dfrac{1}{a(\mu, \sigma)} f(x; \mu, \sigma) & x \geq 0 \end{cases}$$

die Dichtefunktion einer linksseitig bei $x = 0$ gestutzten Normalverteilung mit

$$a(\mu, \sigma) = \int_0^\infty f(x; \mu, \sigma)\, dx\,.$$

Unter Verwendung der Dichte $\varphi(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}$ und der Verteilungsfunktion $\Phi(x) = \int_{-\infty}^{x} \varphi(t)dt$ der standardisierten Normalverteilung gilt auch [16]

$$\begin{aligned} f(x; \mu, \sigma) &= \frac{1}{\sigma}\varphi(z) \qquad \text{mit } z = \frac{x-\mu}{\sigma}\,, \\ a(\mu, \sigma) &= \Phi(v) \qquad \text{mit } v = \frac{\mu}{\sigma}\,, \\ \bar{f}(x; \mu, \sigma) &= \frac{1}{\sigma\Phi(v)}\varphi(z)\,. \end{aligned} \tag{5.6}$$

5.2.2 Anpassung der Reestimierungsformeln

Die linksseitig gestutzte Normalverteilung ist offensichtlich weder elliptisch-symmetrisch, noch ist sie log-konkav im Parameter σ. Folglich muss zunächst geprüft werden, wie sich die in (5.4) definierte und zu maximierende Funktion $\mathcal{Q}_\vartheta(\lambda, \vartheta'_{jm})$ verhält. Dabei können wir unsere Betrachtungen auf ein festes Parameterpaar $\vartheta'_{jm} = (\mu'_{jm}, \sigma'^2_{jm})$ im Zustand j und der Mischkomponente m beschränken und deshalb im Folgenden die Indizes weglassen. Da wir bezüglich ϑ' maximieren wollen, können die Größen ζ_t, in denen die Komponenten der aktuellen Parameter ϑ stecken, einfach als konstante Faktoren betrachtet werden. Der Einfachheit halber bezeichnen wir die zu maximierenden Parameter statt mit ϑ' deshalb im Weiteren mit ϑ und ignorieren den bei gegebenem O und λ festen Faktor $L(O|\lambda)$. Damit entspricht dem Ausdruck in Gleichung (5.4) eine Funktion der Form

$$\mathcal{Q}(\vartheta) = \sum_{t=1}^{T} \zeta_t \log b(O_t; \vartheta)\,, \tag{5.7}$$

bzw. speziell mit der oben eingeführten gestutzten Normalverteilung betrachten wir weiterhin

$$\mathcal{Q}(\mu, \sigma) = \sum_{t=1}^{T} \zeta_t \log \bar{f}(O_t; \mu, \sigma)\,, \qquad \zeta_t, O_t \geq 0\,. \tag{5.8}$$

Ziel ist es nun, die Existenz eines eindeutigen globalen Maximums von (5.8) zu zeigen und dieses Maximum auch zu berechnen. Zunächst gilt folgende Aussage:

Satz 5.1 *Seien $\Omega = \{\mu, \sigma | -\infty < \mu < \infty, \sigma > 0\}$ der Parameterraum der Funktion $\mathcal{Q}(\mu, \sigma)$ aus Gleichung* (5.8) *und $\delta\Omega$ der Rand des Parameterraums. Unter der Annahme, dass unter den O_t mindestens zwei existieren mit $O_{t_i} \neq O_{t_j}$, $\zeta_{t_i} > 0$ und $\zeta_{t_j} > 0$, gilt: $\mathcal{Q}(\mu, \sigma) \to -\infty$ für $(\mu, \sigma) \to \delta\Omega$.*

Beweis: Es gilt mit $z_t = (O_t - \mu)/\sigma$

$$\begin{aligned}\mathcal{Q}(\mu,\sigma) &= \sum_t \zeta_t \log \bar{f}(O_t;\mu,\sigma) \\ &= \sum_t \zeta_t \log\Big(\frac{1}{\sigma\Phi(\mu/\sigma)}\varphi(z_t)\Big) \\ &= \log\prod_t\Big(\frac{\varphi(z_t)}{\sigma\Phi(\mu/\sigma)}\Big)^{\zeta_t}, \end{aligned} \tag{5.9}$$

und damit ist $\lim_{(\mu,\sigma)\to\delta\Omega} \mathcal{Q}(\mu,\sigma) \to -\infty$ genau dann, wenn $\lim_{(\mu,\sigma)\to\delta\Omega} \prod_t \left(\varphi(z_t)/(\sigma\Phi(\mu/\sigma))\right)^{\zeta_t} = 0$.

1. Für $\mu \to \infty$ strebt $\varphi(z_t)$ gegen null und $\Phi(\mu/\sigma)$ gegen 1 für alle σ, folglich ist $\lim_{\mu\to\infty} \varphi(z_t)/(\sigma\Phi(\mu/\sigma)) = 0$.
2. Für $\mu \to -\infty$ streben sowohl alle $\varphi(z_t)$ als auch $\Phi(\mu/\sigma)$ gegen 0. Wir verwenden die für $x > 0$ gültige, konvergente Kettenbruchentwicklung [35]

$$\Phi(x) = 1 - \cfrac{\varphi(x)}{x + \cfrac{1}{x + \cfrac{2}{\cdots}}} .$$

   Damit gilt für $\mu < 0$ und mit $\varphi(-x) = \varphi(x)$

$$\Phi(\mu/\sigma) = 1 - \Phi(-\mu/\sigma) = \cfrac{\varphi(\mu/\sigma)}{-\frac{\mu}{\sigma} + \cfrac{1}{-\frac{\mu}{\sigma} + \cfrac{2}{\cdots}}} ,$$

   und daraus folgt

$$\begin{aligned}\prod_t\left(\frac{\varphi(z_t)}{\sigma\Phi(\mu/\sigma)}\right)^{\zeta_t} &= \left(-\frac{\mu}{\sigma^2} + \cfrac{1}{-\mu + \cfrac{2}{-\frac{\mu}{\sigma^2} + \cfrac{3}{\cdots}}}\right)^{\sum_t \zeta_t} \prod_t\left(\frac{\varphi(z_t)}{\varphi(\mu/\sigma)}\right)^{\zeta_t} \\ &= \left(-\frac{\mu}{\sigma^2} + \cdots\right)^{\sum_t \zeta_t} e^{-\frac{1}{2\sigma^2}\sum_t \zeta_t (O_t^2 - 2\mu O_t)} . \end{aligned} \tag{5.10}$$

   Da nach Voraussetzung mindestens zwei verschiedene $O_t \geq 0$ mit $\zeta_t > 0$ existieren, kann die Summe im Exponenten für negative μ nicht null werden. Schließlich ist $\lim_{x\to\infty} x^b e^{-ax} = 0$ für alle $a, b > 0$, und somit geht der ganze Ausdruck (5.10) für $\mu \to -\infty$ gegen 0.
3. Für $\sigma \to \infty$ strebt $\Phi(\mu/\sigma)$ gegen $1/2$ und $\varphi(z_t)$ gegen $\varphi(0)$ für alle μ, somit gilt $\lim_{\sigma\to\infty} \varphi(z_t)/(\sigma\Phi(\mu/\sigma)) = 0$.

4. Für $\sigma \to 0$ strebt $\Phi(\mu/\sigma)$ gegen 1, falls $\mu > 0$, und gegen 0.5 für $\mu = 0$. In beiden Fällen ist

$$\begin{aligned}\prod_t \Big(\frac{\varphi(z_t)}{\sigma\Phi(\mu/\sigma)}\Big)^{\zeta_t} &= \Big(\frac{1}{\Phi(\mu/\sigma)}\Big)^{\sum_t \zeta_t}\Big(\frac{1}{\sigma}\Big)^{\sum_t \zeta_t}\prod_t \varphi(z_t)^{\zeta_t} \\ &= \Big(\frac{1}{\Phi(\mu/\sigma)}\Big)^{\sum_t \zeta_t}\Big(\frac{1}{\sigma}\Big)^{\sum_t \zeta_t}\frac{1}{\sqrt{2\pi}}\, e^{-\frac{1}{2\sigma^2}\sum_t \zeta_t (O_t-\mu)^2}\,. \qquad (5.11)\end{aligned}$$

Wieder nach Voraussetzung kann die Summe im Exponenten nicht null werden und damit strebt (5.11) für $\sigma \to 0$ bei $\mu \geq 0$ gegen 0.
Sei nun $\mu < 0$. Dann geht für $\sigma \to 0$ auch $\Phi(\mu/\sigma)$ gegen 0. Hier gilt aber wieder die Gleichung (5.10) aus Punkt 2, deren Grenzwert mit den gleichen Argumenten wie oben auch für $\sigma \to 0$ und festem $\mu < 0$ gleich 0 ist.

□

Nach Satz 5.1 nimmt die Q-Funktion (5.8) im Innern von Ω ihr globales Maximum an, das jedoch nicht eindeutig sein muss. Zur Bestimmung der Maximalstellen betrachten wir nun die kritischen Punkte der Q-Funktion, die wir durch Nullsetzen der partiellen Ableitungen nach μ und σ erhalten.

Zunächst bestimmen wir dazu die partiellen Ableitungen der Dichte $\bar{f}(x;\mu,\sigma)$. Der Einfachheit halber bezeichne $f(\mu) = f(x;\mu,\sigma)$ die oben eingeführte Dichtefunktion bei konstanten x und σ und entsprechend sind die Funktionen $a(\mu)$, $\bar{f}(\mu)$, $f(\sigma)$, $a(\sigma)$ und $\bar{f}(\sigma)$ zu verstehen. Dann gilt mit $\Phi'(y) = \varphi(y)$, $\varphi'(y) = -y\varphi(y)$, $\frac{\partial z}{\partial \mu} = -\frac{\partial v}{\partial \mu} = -\frac{1}{\sigma}$, $\frac{\partial z}{\partial \sigma} = -\frac{1}{\sigma}z$ und $\frac{\partial v}{\partial \sigma} = -\frac{1}{\sigma}v$:

$$\begin{aligned} f'(\mu) &= \frac{1}{\sigma}\varphi'(z)\frac{\partial z}{\partial \mu} \\ &= f(\mu)\frac{x-\mu}{\sigma^2}\,, \\ a'(\mu) &= \frac{1}{\sigma}\varphi(v) \\ &= f(0;\mu,\sigma)\,, \\ \bar{f}'(\mu) &= \frac{1}{a(\mu)}f'(\mu) - \frac{1}{a^2(\mu)}f(\mu)a'(\mu) \\ &= \bar{f}(\mu)\frac{x-\mu}{\sigma^2} - \bar{f}(\mu)\frac{1}{a(\mu)}f(0;\mu,\sigma) \\ &= \bar{f}(\mu)\left[\frac{x-\mu}{\sigma^2} - \bar{f}(0;\mu,\sigma)\right]\,, \qquad (5.12) \\ f'(\sigma) &= -\frac{1}{\sigma^2}\varphi(z) + \frac{1}{\sigma}\varphi'(z)\frac{\partial z}{\partial \sigma} \\ &= -\frac{1}{\sigma^2}\varphi(z) + \frac{1}{\sigma^2}z^2\varphi(z) \\ &= f(\sigma)\left[\frac{(x-\mu)^2}{\sigma^3} - \frac{1}{\sigma}\right]\,, \end{aligned}$$

$$
\begin{aligned}
a'(\sigma) &= -\frac{1}{\sigma} v\varphi(v) \\
&= -\frac{\mu}{\sigma} f(0;\mu,\sigma)\,, \\
\bar{f}'(\sigma) &= \frac{1}{a(\sigma)} f'(\sigma) - \frac{1}{a^2(\sigma)} f(\mu) a'(\sigma) \\
&= \bar{f}(\sigma)\left[\frac{(x-\mu)^2}{\sigma^3} - \frac{1}{\sigma}\right] + \bar{f}(\sigma)\frac{\mu}{\sigma}\bar{f}(0;\mu,\sigma) \\
&= \bar{f}(\sigma)\left[\frac{(x-\mu)^2}{\sigma^3} - \frac{1}{\sigma} + \frac{\mu}{\sigma}\bar{f}(0;\mu,\sigma)\right]\,. \qquad (5.13)
\end{aligned}
$$

Mit Hilfe dieser Gleichungen erhalten wir die partiellen Ableitungen $\partial\mathcal{Q}/\partial\mu$ und $\partial\mathcal{Q}/\partial\sigma$:

$$
\begin{aligned}
\frac{\partial\mathcal{Q}}{\partial\mu} &= \sum_t \zeta_t \frac{\bar{f}'(\mu)}{\bar{f}(\mu)} \\
&= \sum_t \zeta_t \left[\frac{O_t-\mu}{\sigma^2} - \bar{f}(0;\mu,\sigma)\right]\,, \qquad (5.14) \\
\frac{\partial\mathcal{Q}}{\partial\sigma} &= \sum_t \zeta_t \frac{\bar{f}'(\sigma)}{f(\sigma)} \\
&= \sum_t \zeta_t \left[\frac{(O_t-\mu)^2}{\sigma^3} - \frac{1}{\sigma} + \frac{\mu}{\sigma}\bar{f}(0;\mu,\sigma)\right]\,. \qquad (5.15)
\end{aligned}
$$

Die kritischen Punkte $(\bar{\mu},\bar{\sigma})$ der Q-Funktion ergeben sich aus $\partial\mathcal{Q}/\partial\mu = \partial\mathcal{Q}/\partial\sigma = 0$ und führen zu zwei impliziten Gleichungen:

$$
\begin{aligned}
\frac{\sum_t \zeta_t O_t}{\sum_t \zeta_t} - \bar{\mu} - \bar{\sigma}^2\,\bar{f}(0;\bar{\mu},\bar{\sigma}) &= 0\,, \qquad (5.16) \\
\frac{\sum_t \zeta_t (O_t-\bar{\mu})^2}{\sum_t \zeta_t} - \bar{\sigma}^2 + \bar{\mu}\bar{\sigma}^2\,\bar{f}(0;\bar{\mu},\bar{\sigma}) &= 0\,. \qquad (5.17)
\end{aligned}
$$

Dieses Gleichungssystem lässt sich analytisch nicht explizit nach $\bar{\mu}$ und $\bar{\sigma}$ auflösen, kann jedoch auf eine eindimensionale implizite Gleichung für $\bar{\mu}$ reduziert werden. Wir setzen

$$
\begin{aligned}
E &:= \frac{\sum_t \zeta_t O_t}{\sum_t \zeta_t}\,, \qquad (5.18) \\
V &:= \frac{\sum_t \zeta_t O_t^2}{\sum_t \zeta_t}\,, \qquad (5.19) \\
\bar{f}_0(\mu,\sigma) &:= \bar{f}(0;\mu,\sigma)\,, \qquad (5.20)
\end{aligned}
$$

und damit wird aus (5.16) und (5.17)

$$
\begin{aligned}
E - \bar{\mu} - \bar{\sigma}^2\,\bar{f}_0(\bar{\mu},\bar{\sigma}) &= 0\,, \qquad (5.21) \\
V - 2\bar{\mu}E + \bar{\mu}^2 - \bar{\sigma}^2 + \bar{\mu}\bar{\sigma}^2\,\bar{f}_0(\bar{\mu},\bar{\sigma}) &= 0\,. \qquad (5.22)
\end{aligned}
$$

Aus (5.21) folgt

$$\bar{\sigma}^2 \, \bar{f}_0(\bar{\mu}, \bar{\sigma}) = E - \bar{\mu} \,,$$

und das eingesetzt in (5.22) ergibt

$$\begin{aligned} \bar{\sigma}^2 &= V - 2\bar{\mu}E + \bar{\mu}^2 + \bar{\mu}E - \bar{\mu}^2 \\ &= V - \bar{\mu}E \,. \end{aligned} \tag{5.23}$$

Dies wiederum eingesetzt in (5.21) führt zu der impliziten Gleichung, in der nur noch $\bar{\mu}$ enthalten ist:

$$p(\bar{\mu}) := E - \bar{\mu} - (V - \bar{\mu}E) \, \bar{f}_0(\bar{\mu}, \sqrt{V - \bar{\mu}E}) = 0 \,. \tag{5.24}$$

Die Nullstellen der Funktion $p(\mu)$ bilden folglich zusammen mit den jeweils daraus nach (5.23) berechneten σ die kritischen Punkte der Funktion $\mathcal{Q}(\mu, \sigma)$, von denen einer dem gesuchten globalen Maximum entspricht.

Da sich die Gleichung $p(\mu) = 0$ nicht analytisch nach μ auflösen lässt, wird an dieser Stelle bereits klar, dass wir für ein HMM mit gestutzten Normalverteilungen keine expliziten Reestimierungsformel für die Parameter μ aufstellen können. Allerdings werden wir im nächsten Abschnitt sehen, dass die einzige Nullstelle von $p(\mu)$ unter bestimmten Bedingungen numerisch berechnet werden kann.

5.2.3 Numerische Berechnung der Reestimierungsparameter $\bar{\mu}$ und $\bar{\sigma}^2$

Nach den bisherigen Überlegungen liegt der Schlüssel zur Maximierung der Q-Funktion bezüglich der Parameter (μ, σ) in der Berechnung der Nullstellen der Funktion $p(\mu)$.

Korollar 5.2 *Unter den Annahmen aus Satz 5.1 existiert mindestens ein Punkt μ' in $\Omega_p = \{\mu | -\infty < \mu < V/E\}$ mit $p(\mu') = 0$.*

Beweis: Aus Satz 5.1 folgt, dass es es mindestens einen Punkt (μ', σ') im Innern von $\Omega = \{\mu, \sigma | -\infty < \mu < \infty, \sigma > 0\}$ geben muss, an dem das globale Maximum der Q-Funktion (5.8) angenommen wird. An dieser Stelle muss notwendigerweise $\partial\mathcal{Q}/\partial\mu = \partial\mathcal{Q}/\partial\sigma = 0$ gelten, und damit gilt nach Konstruktion $p(\mu') = 0$ (5.23) und $\sigma' = \sqrt{V - \mu' E}$ (5.24). Da σ' im Innern von Ω strikt positiv ist, folgt für den Extremalpunkt: $\mu' < V/E$.

□

Als Nächstes stellt sich die Frage nach der Anzahl der möglichen Nullstellen von $p(\mu)$. Falls es nur eine einzige Nullstelle gibt, muss diese dem globalen Maximum der Q-Funktion entsprechen. Der nichtlineare und analytisch nicht auflösbare Term $\Phi(\frac{\mu}{\sqrt{V-\mu E}})$, der in in $p(\mu)$ bzw. $\bar{f}_0$ steckt, macht jedoch eine analytische Untersuchung der Funktion extrem schwierig. Daneben können die von den Ausgaben abhängigen Parameter E und V theoretisch jeden Wert > 0 annehmen. Wir beschränken uns deshalb an dieser Stelle auf eine graphische Untersuchung von

$p(\mu)$. Dazu klären wir zuerst, wie die Werte E und V zusammenhängen und zeigen dann, dass wir die Kurvendiskussion auf ein festes, aber beliebiges E beschränken können.

Nach Definition sind $E, V \geq 0$, da wir bei Verwendung der gestutzten Normalverteilung nur Daten mit $O_t \geq 0$ voraussetzen. Unter der Annahme aus Satz 5.1 – es existieren mindestens zwei verschiedene O_t mit Gewichten $\zeta_t > 0$ – , die wir im Weiteren implizieren, sind E und V damit immer positiv. Mit $\gamma_t := \zeta_t / \sum_t \zeta_t$, $t \in \{1, \ldots, T\}$ ist $\sum_t \gamma_t = 1$, und aus der Konvexität einer quadratischen Funktion folgt

$$V - E^2 = \sum_t \gamma_t O_t^2 - \left(\sum_t \gamma_t O_t \right)^2 \geq 0 \,. \tag{5.25}$$

Wiederum mit der Annahme aus Satz 5.1 gilt sogar $V > E^2$.

Lemma 5.3 *Für ein festes $\bar{E} > 0$ und jedes beliebige $\bar{V} > \bar{E}^2$ gelte: $p(\mu; \bar{E}, \bar{V})$ besitzt eine einzige und eindeutige Nullstelle in $\bar{\mu}$. Dann gilt für beliebige $E > 0$ und $V > E^2$: $p(\mu; E, V)$ besitzt nur eine einzige Nullstelle in $\mu' = (E/\bar{E})\bar{\mu}$.*

Beweis: Sei μ' eine Nullstelle von $p(\mu; E, V)$, dann gilt mit der Schreibweise von Gleichung (5.6) und mit $\kappa := \bar{E}/E$:

$$\begin{aligned}
0 &= \kappa \cdot p(\mu'; E, V) \\
&= \kappa E - \kappa\mu' - \kappa\sqrt{V - \mu' E}\, \frac{\varphi(\frac{\mu'}{\sqrt{V-\mu' E}})}{\Phi(\frac{\mu'}{\sqrt{V-\mu' E}})} \\
&= \kappa E - \kappa\mu' - \sqrt{\kappa^2 V - \kappa\mu'\kappa E}\, \frac{\varphi(\frac{\kappa\mu'}{\sqrt{\kappa^2 V-\kappa\mu'\kappa E}})}{\Phi(\frac{\kappa\mu'}{\sqrt{\kappa^2 V-\kappa\mu'\kappa E}})} \\
&= p(\kappa\mu'; \kappa E, \kappa^2 V) \\
&= p(\kappa\mu'; \bar{E}, \kappa^2 V) \,.
\end{aligned} \tag{5.26}$$

Folglich ist $\kappa\mu'$ Nullstelle von $p(\mu; \bar{E}, \kappa^2 V)$ und es gilt $\bar{V} := \kappa^2 V > \bar{E}^2$. Nach Voraussetzung ist diese Nullstelle aber eindeutig, d. h. $\kappa\mu' = \bar{\mu}$, und somit ist die Nullstelle $\mu' = (E/\bar{E})\bar{\mu}$ für $p(\mu; E, V)$ ebenfalls eindeutig.

□

Nach Lemma 5.3 genügt es, die Funktion $p(\mu; E, V)$ für ein festes E und bei variierendem V auf ihre Nullstellen zu untersuchen. In Abbildung 5.2 sind die Kurven der Funktion für drei verschiedene Parameter V bei festem E aufgetragen. Der Verlauf dieser Kurven lässt vermuten, dass p generell nur eine einzige Nullstelle besitzt. Nach einer Begutachtung von zahlreichen Kurven $p(\mu; E, V)$ bei variierendem V und mit Berücksichtigung der Stetigkeit von p, die ein unerwartetes Verhalten der Kurve für die nichtbetrachteten Parameterwerte extrem unwahrscheinlich macht, können wir davon ausgehen, dass eine numerisch bestimmte Nullstelle der Funktion p eindeutig ist und dem gesuchten globalen Maximum der Q-Funktion $\mathcal{Q}(\mu, \sigma)$ entspricht. Ein analytischer Beweis dazu konnte allerdings aufgrund der angesprochenen Schwierigkeiten nicht aufgestellt werden.

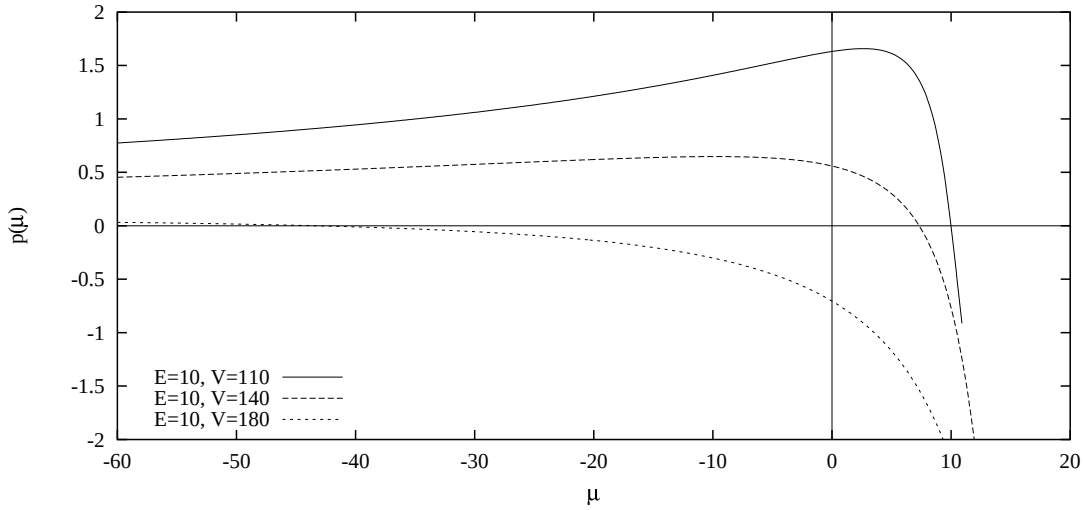


Abbildung 5.2: $p(\mu; E, V)$ mit $E = 10$ und verschiedenen Parametern V

Abbildung 5.2 macht auch eine Verschiebung der Nullstelle nach links deutlich, wenn V gegenüber E wächst. Für $V = \frac{\pi}{2}E^2$ liegt die Nullstelle exakt bei $\mu = 0$, was sich sofort aus der Gleichung $p(0) = 0$ ergibt, und für noch größere V wandert sie in den negativen Bereich. Da in diesem Bereich $p(\mu)$ sehr flach ist, strebt die Nullstelle für wachsende V sehr schnell gegen $-\infty$. Insofern ist bei sehr großen Werten für V gegenüber E mit numerischen Problemen bei der Nullstellenbestimmung zu rechnen.

Zur numerische Berechnung der Nullstelle von $p(\mu)$ verwenden wir den Algorithmus von Brent (siehe [37], S. 359). Dieser stellt ein Verfahren zur Nullstellensuche bei eindimensionalen Funktionen dar und kombiniert im Wesentlichen Bisektionsverfahren in geeigneter Weise mit Methoden höherer Ordnung. Bei Brents Verfahren werden keine Ableitungen der jeweiligen Funktion berechnet, was uns angesichts der komplizierten Form und der aufwendig zu berechnenden Ableitung von $p(\mu)$ als am geeignetsten erschien; grundsätzlich wären aber auch Verfahren einsetzbar, die Ableitungen von $p(\mu)$ berücksichtigen. Dem Algorithmus von Brent muss allerdings ein Intervall vorgegeben werden, in dem die Nullstelle sicher liegt. Wir müssen deshalb vorab eine linke und eine rechte Grenze für die gesuchte Nullstelle $\bar{\mu}$ berechnen:

Aus $p(\mu) = 0$ erhalten wir zunächst

$$\bar{\mu} = E - (V - \mu E)\, \bar{f}_0(\mu, \sqrt{V - \mu E})\,,$$

und daraus folgt

$$\mu < E\,, \tag{5.27}$$

da $(V - \mu E)$ für den Definitionsbereich $\mu < V/E$ immer positiv ist und $\bar{f}_0$ als Dichtefunktion ebenfalls nur positive Werte annimmt.

Die linke Grenze ist etwas schwieriger zu bestimmen, da die Nullstelle auf der μ-Achse beliebig weit im negativen Bereich liegen kann. Allerdings gibt es bei der Berechnung der Funktion

$1/\Phi(x)$ eine numerische Grenze, bei der $\Phi(x)$ null wird. Dieser Punkt lässt sich genau bestimmen, da wir die Funktionswerte der analytisch nicht berechenbaren Funktion $\Phi(x)$ im Programm einmal vorweg an äquidistanten Stützstellen x_s einlesen und dann linear interpolieren. Wir setzen

$$C := \min\{x_s | \Phi(x_s) > 0\} . \tag{5.28}$$

$C < 0$ entspricht damit der kleinsten Stützsstelle x_s, für die $1/\Phi(x_s)$ berechnet werden kann. Für ein festes μ muss zur Auswertung der Funktion $p(\mu)$ die Dichte $f_0(\mu, \sqrt{V - \mu E})$ berechnet werden, in die wiederum $\Phi(\mu/\sqrt{V - \mu E})$ eingeht (Gleichungen (5.24), (5.20) und (5.6)). Daraus ergibt sich für $\mu < 0$ die Bedingung

$$\begin{array}{rrcl} & \mu & \geq & C\sqrt{V - \mu E} \\ \Longleftrightarrow & \mu^2 & \leq & C^2(V - \mu E) \\ \Longleftrightarrow & \mu^2 + C^2 E\mu & \leq & C^2 V \\ \Longleftrightarrow & (\mu + \frac{C^2 E}{2})^2 - \frac{C^4 E^2}{4} & \leq & C^2 V \\ \Longleftrightarrow & \pm\left(\mu + \frac{C^2 E}{2}\right) & \leq & \pm C\sqrt{V + \frac{C^2 E^2}{4}} \\ \Longleftrightarrow & \cdots \geq \mu & \geq & C\sqrt{V + \frac{C^2 E^2}{4}} - \frac{C^2 E}{2} . \end{array} \tag{5.29}$$

Zusammen mit der bereits festgestellten Tatsache, dass $p(0; E, \frac{\pi}{2}E^2) = 0$ ist, gelangen wir somit zu folgenden Einschließungen μ_L und μ_R für die Nullstelle von $p(\mu; E, V)$:

$$\mu_L = \begin{cases} C\sqrt{V + \frac{C^2 E^2}{4}} - \frac{C^2 E}{2} & V > \frac{\pi}{2}E^2 \\ 0 & V \leq \frac{\pi}{2}E^2 \end{cases} , \tag{5.30}$$

$$\mu_R = \begin{cases} 0 & V > \frac{\pi}{2}E^2 \\ E & V \leq \frac{\pi}{2}E^2 \end{cases} . \tag{5.31}$$

Das so programmierte Verfahren liefert z. B. bei einer Genauigkeit von 10^{-6} nach durchschnittlich weniger als 10 und maximal 20 Schritten die gesuchte Nullstelle, wobei pro Schritt einmal der Funktionswert von p berechnet werden muss. Schließlich sei noch bemerkt, dass die linke Grenze für $p(\mu; E, V)$ multipliziert mit einem Faktor $\kappa > 0$ wiederum die linke Grenze für $p(\kappa\mu; \kappa E, \kappa^2 V)$ darstellt, so dass μ_L bei größerem E und entsprechendem V mit dem gleichen Faktor nach rechts wandert wie die Nullstelle (vgl. Lemma 5.3).

Die von der Konstanten C abhängige linke Grenze für μ stellt zwar die Berechenbarkeit der Funktion $p(\mu)$ für alle $\mu \geq \mu_L$ sicher, es treten jedoch schon nahe dieser Grenze numerische Instabilitäten auf, die aus der numerischen Berechnung für $\Phi(x)$ resultieren: In Abbildung 5.3 ist zu sehen, wie sich für sehr kleine μ bei der Division der zwei extrem kleinen Funktionen φ und Φ starke oszillierende Effekte ergeben. Das Problem besteht darin, dass in diesem Bereich von μ die Funktion Φ bei der üblichen Genauigkeit von 16 Nachkommastellen über mehrere Stützstellen hinweg den gleichen Funktionswert annimmt, so dass selbst bei einfacher Interpolation quasi eine Treppenfunktion entsteht.

Das Problem lässt sich im Wesentlichen entschärfen, indem wir die Werte für die Funktion $\Phi(x)$ an negativen Stützstellen x_s berechnen. Die Φ-Werte liegen dann zwischen 0 und 0.5 und können

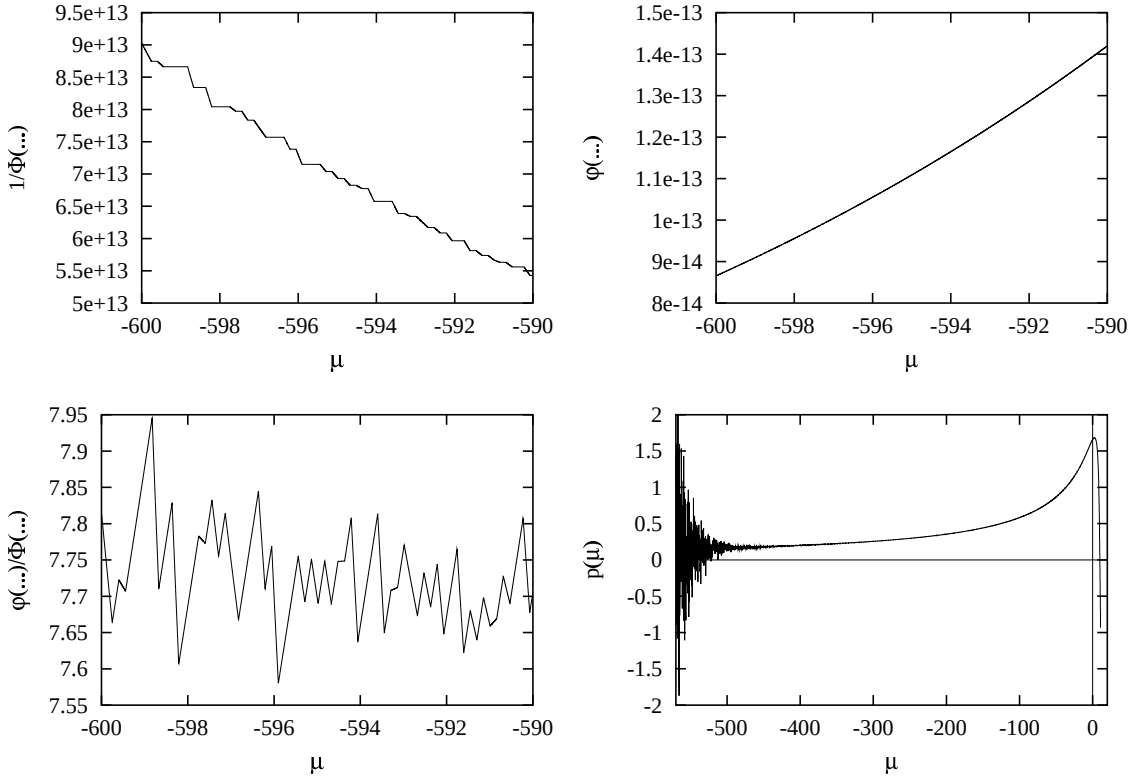


Abbildung 5.3: Numerische Instabilität bei der Berechnung von $p(\mu)$

somit im kritischen Bereich um 0 durch Gleitkommadarstellung auf bis zu 50 Nachkommastellen genau dargestellt werden (die Φ-Werte werden mit dem Computeralgebra-Programm MATHEMATICA einmal vorab numerisch berechnet). Dadurch wird auch die Konstante C aus Gleichung (5.28) wesentlich kleiner und das Intervall zur Nullstellensuche wird auf der linken Seite vergrößert. Allerdings treten auch dann noch Oszillationen am Rand μ_L auf. Als weitere Maßnahme interpolieren wir deshalb nicht die Werte von $\Phi(x)$, sondern die der Funktion $p(\mu)$ selbst:

1. Setze $x = \mu/\sqrt{V - \mu E}$
2. Bestimme Stützstellen x_s, x_{s+1} mit $x_s \leq x \leq x_{s+1}$
3. Berechne korrespondierende $\mu_1 = x_s\sqrt{V + \frac{x_s^2 E^2}{4}} - \frac{x_s^2 E}{2}$ und μ_2 analog mit x_{s+1}
4. Interpoliere $p(\mu)$ linear zwischen $p(\mu_1)$ und $p(\mu_2)$

Die Gleichung für μ_1 und μ_2 entspricht einer Umkehrung der Gleichung im ersten Schritt und der Berechnung von μ_L nach (5.29). Dabei liegt wieder die Tatsache zugrunde, dass in die Auswertung von $p(\mu)$ $\Phi(\mu/\sqrt{V - \mu E})$ eingeht.

Trotz der hohen Genauigkeit der Φ-Werte kann es vorkommen, dass für $V \gg E^2$ die Nullstelle der Funktion $p(\mu)$ links vom numerischen Berechnungsintervall $\mathcal{I} = [\mu_L, \mu_R]$ liegt, d. h. $p(\mu) < 0$ in ganz $\mathcal{I}$. Das heißt aber für die Q-Funktion, deren Steigung proportional zu $p(\mu)$ ist (mit

gleichem Vorzeichen), dass ihr Maximum auf dem Rand μ_L liegt. Somit setzen wir die gesuchte Maximalstelle auf $\overline{\mu} = \mu_L$. Es hat sich allerdings bei den Anwendungen gezeigt, dass dieser Fall in den Daten sehr selten auftritt. Bei genauerer Betrachtung der Definitionen (5.18) und (5.19) lässt sich dies leicht erklären: E entspricht dem statistischen Mittelwert der gewichteten Daten und damit berechnet sich die gewichtete statistische Varianz der Daten zu

$$\frac{\sum_t \zeta_t (O_t - E)^2}{\sum_t \zeta_t} = V - E^2 \,. \tag{5.32}$$

Wenn z. B. bei identischen Gewichten ζ_t für die eine Hälfte der Daten $O_t = 0$ gilt – d. h. sie liegen auf der linken Grenze der gestutzten Verteilung – und für die andere Hälfte $O_t = 2E$, ist $V = 2E^2$. Diese Konstellation stellt bereits einen Extremfall dar, der bei ausreichend vielen Daten statistisch unwahrscheinlich wird, zumal die Modellparameter im Laufe des Reestimierungsalgorithmus immer besser an die Daten angepasst werden.

Abschließend fassen wir zusammen: Die Berechnung der neuen Dichteparameter $(\overline{\mu}_{jk}, \overline{\sigma}^2_{jk})$ für jeden Zustand j und jede Mischkomponente k bei Verwendung der in Abschnitt 5.2.1 definierten gestutzten Normalverteilung erfolgt in folgenden Schritten:

1. Berechnung der Größen $\zeta_t(j,k)$ nach (2.24c) für alle t, und damit Berechnung von E, V nach (5.18) und (5.19)
2. Berechnung der Grenzen μ_L, μ_R nach (5.30) und (5.31)
3. Numerische Nullstellenbestimmung von $p(\mu)$ (5.24) in den Grenzen μ_L, μ_R ergibt $\overline{\mu}_{jk}$
4. $\overline{\sigma}^2_{jk} = V - \overline{\mu}_{jk} E$ nach (5.23)

Der Vorteil der Berechnung der Parameter $(\overline{\mu}_{jk}, \overline{\sigma}^2_{jk})$ liegt in der von der iterativen, numerischen Nullstellensuche getrennten Bestimmung der Größen E und V anhand der Daten. Dadurch wird die Laufzeit des gesamten Verfahrens nur um einen konstanten Wert erhöht, der vor allem nicht von der Länge der Daten oder den Modellparametern abhängt.

Die bisher betrachteten Formeln und Gleichungen gelten jeweils nur für das Training mit einer einzigen Sequenz. Eine Erweiterung auf mehrere Sequenzen erfolgt jedoch auch hier wieder problemlos (vgl. Abschnitt 2.5). Die Q-Funktion (5.3) erweitert sich dann um die Summe über alle Sequenzen. Damit läuft auch die spezielle Q-Funktion $\mathcal{Q}(\mu, \sigma)$ für die gestutzte Normalverteilung aus Gleichung (5.8) über alle Sequenzen, und im Endeffekt ändert sich nur die Berechnung der Größen E und V nach (5.18) und (5.19), bei denen dann in Zähler und Nenner jeweils genauso über alle Sequenzen summiert werden muss.

5.2.4 Konsistenz mit Erwartungswert und Varianz

Zum besseren Verständnis der impliziten Gleichungen (5.21) und (5.22) für die neuen Parameter $(\overline{\mu}, \overline{\sigma}^2)$ vergleichen wir diese Gleichungen mit den expliziten Reestimierungsformeln (2.25d) und (2.25e) bei Einsatz der nichtgestutzten Normalverteilung, die ja unter die Klasse der elliptisch-symmetrischen Verteilungen fällt (vgl. Abschnitt 2.3). Bei Verwendung der nach (5.18) und (5.19) definierten Größen E, V gelten dort die Gleichungen $\overline{\mu} = E$ und $\overline{\sigma}^2 = V - E^2$. Nach

den bisherigen Überlegungen bedeutet das aber, dass die Q-Funktion genau dann maximal wird, wenn die Parameter der Dichtefunktion so gesetzt werden, dass der Erwartungswert der Verteilung gleich dem statistischen Mittelwert und die Varianz der Verteilung gleich der statistischen Varianz der gewichteten Daten ist (Gleichungen (5.18) und (5.32)). Dabei ist das Gewicht ζ_t proportional zur Wahrscheinlichkeit, dass sich die Sequenz bei Ausgabe des Zeichens O_t im jeweils betrachteten Zustand und der jeweiligen Mischkomponente befindet (vgl. (5.4) in Abschnitt 5.1).

Es stellt sich nun die Frage, ob das bei der gestutzten Normalverteilung genauso gilt. Erwartungswert und Varianz der nach der Dichte $f(x;\mu,\sigma)$ verteilten Zufallsvariablen X lassen sich folgendermaßen berechnen [16, 19]:

$$EX = \mu + \sigma^2 \bar{f}_0(\mu,\sigma)\,, \tag{5.33}$$

$$VarX = \sigma^2 - \mu\sigma^2 \bar{f}_0(\mu,\sigma) - \sigma^4 \bar{f}_0^2(\mu,\sigma)\,. \tag{5.34}$$

Ein Vergleich von (5.33) mit (5.21) zeigt sofort, dass auch hier der statistische Mittelwert mit dem analytischen Erwartungswert der Verteilung zusammenfällt, wenn μ und σ die Q-Funktion maximieren, d. h. $E = EX$. Wie sieht es aber mit der Varianz aus? Unter der Annahme, dass statistische und analytische Varianz identisch sind, gilt nach (5.34)

$$V - E^2 = \sigma^2 - \mu\sigma^2 \bar{f}_0(\mu,\sigma) - \sigma^4 \bar{f}_0^2(\mu,\sigma)\,.$$

Mit $E = EX$ eingesetzt in (5.33) gilt aber auch

$$V - E^2 = V - \mu^2 - 2\mu\sigma^2 \bar{f}_0(\mu,\sigma) - \sigma^4 \bar{f}_0^2(\mu,\sigma)\,,$$

und somit ergibt sich nach Gleichsetzen und Kürzen

$$V = \mu^2 + \sigma^2 + \mu\sigma^2 \bar{f}_0(\mu,\sigma)\,.$$

Diese Gleichung erweitern wir auf beiden Seiten mit $-2\mu E + \mu^2$ und erhalten

$$\begin{aligned} V - 2\bar{\mu}E + \bar{\mu}^2 &= \mu^2 + \sigma^2 + \mu\sigma^2 \bar{f}_0(\mu,\sigma) - 2\bar{\mu}E + \bar{\mu}^2 \\ &= \mu^2 + \sigma^2 + \mu\sigma^2 \bar{f}_0(\mu,\sigma) - 2\mu^2 - 2\mu\sigma^2 \bar{f}_0(\mu,\sigma) + \bar{\mu}^2 \\ &= \sigma^2 - \mu\sigma^2 \bar{f}_0(\mu,\sigma)\,. \end{aligned}$$

Dies entspricht aber genau der Gleichung (5.22), die sich aus der Bedingung $\partial\mathcal{Q}/\partial\sigma = 0$ zur Maximierung der Q-Funktion ergab.

Als Ergebnis halten wir fest, dass auch bei der gestutzten Normalverteilung in jedem Reestimierungsschritt die Parameter der Dichtefunktion so gesetzt werden, dass Erwartungswert und Varianz der daraus entstehenden Verteilung gleich dem statistischen Erwartungswert und der statistischen Varianz der gewichteten Daten sind.

5.2.5 Praktische Probleme und Modifikationen

In Satz 5.1 und Korollar 5.2 wurde jeweils vorausgesetzt, dass mindestens zwei verschiedene Daten mit positiven Gewichten in jede Parameterbestimmung eingehen, so dass immer $E, V > 0$

und $V > E^2$ gilt. Bei praktischen Anwendungen möchten wir aber auch gerne die Fälle abfangen, in denen alle Daten den gleichen Wert annehmen. Gerade bei unseren Zeitreihen von mitunter sehr regelmäßig einzahlenden Sparern kann es durchaus zu solchen Extremfällen kommen.

Für $V = E^2 > 0$ liegt die rechte Definitionsgrenze von $p(\mu)$ bei $\mu = V/E = E$ (vgl. Korollar 5.2), und es zeigt sich, dass $p(\mu) > 0$ für alle $\mu < E$, obgleich nach Anhang B der rechte Grenzwert $\lim_{\mu \to E} p(\mu) = E - V/E = 0$ auf eine approximierte Lösung hoffen läßt (vgl. Abbildung 5.4). Für $V = E^2 = 0$ ist $p(\mu)$ jedoch überhaupt nicht definiert.

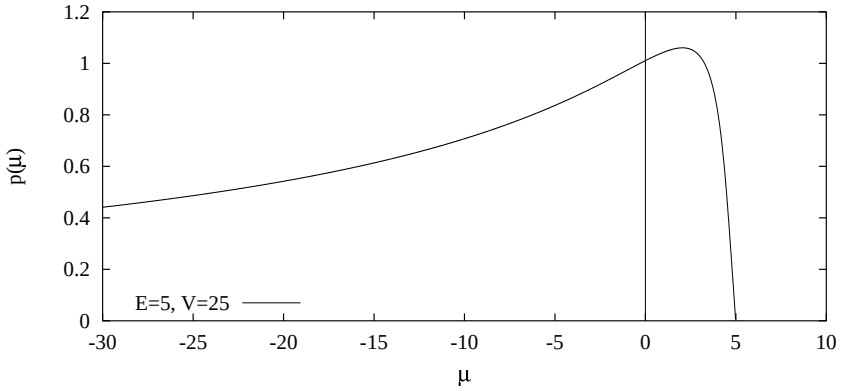


Abbildung 5.4: $p(\mu; E, V)$ mit $E = \sqrt{V} = 5$

Die Sachlage wird klarer, wenn wir uns die ursprüngliche Q-Funktion (5.8) anschauen. $V = E^2$ bedeutet nach (5.25) und wegen der strengen Konvexität der Funktion y^2, dass alle O_t den gleichen Wert $x \geq 0$ annehmen und damit auch $V = E^2 = x^2$ gilt. Folglich vereinfacht sich die Q-Funktion zu

$$\mathcal{Q}(\mu, \sigma) = G \cdot \log \bar{f}(x; \mu, \sigma) \qquad \text{mit } G = \sum_t \zeta_t \,,$$

d. h. eine Maximierung der Q-Funktion ist äquivalent zur Maximierung von $\bar{f}(x; \mu, \sigma)$. Unter der Voraussetzung $x > 0$ wird die Dichte der gestutzten Normalverteilung aber genau dann unendlich groß, wenn $\mu = x$ und $\sigma \to 0$. Dies geht aus dem Beweis von Satz 5.1 hervor, da dann im Punkt 4. die Summe im Exponenten gleich 0 wird, während sowohl für $\mu < 0$ und $\sigma \to 0$ als auch für festes σ und $\mu \to -\infty$ die Dichte $\bar{f}(x; \mu, \sigma) = \frac{1}{\sigma \Phi(\mu/\sigma)} \varphi(\frac{x-\mu}{\sigma})$ gegen 0 strebt (vgl. Gleichung (5.10) in Punkt 2. mit $O_t = x > 0$). Bei Beschränkung des Parameterraums auf $\sigma \geq \sigma_{\min} > 0$ bedeutet das aber, dass $\bar{f}(x; \mu, \sigma)$ und somit auch die Q-Funktion ihr Maximum auf dem Rand $\sigma_{\min}$ annehmen. Für festes σ und $x > 0$ hat $\bar{f}(x; \mu, \sigma)$ ein eindeutiges globales Maximum bezüglich μ, das wiederum durch Nullsetzen der Ableitung (5.12) bestimmt werden kann: Wir erhalten die numerisch lösbare Gleichung für $\bar{\mu}$

$$x - \bar{\mu} - \sigma_{\min}^2 \, \bar{f}_0(\bar{\mu}, \sigma_{\min}) = 0 \,,$$

die mit $x = E$ zwangsläufig identisch ist mit (5.21).

Sei nun $V = E^2 = 0$ und damit auch $x = 0$. Zwar wird auch hier die Dichte der gestutzten Normalverteilung unendlich groß, wenn $\mu = x$ und $\sigma \to 0$; allerding gilt das jetzt auch genauso für alle $\mu < 0$ und sogar für beliebige, feste σ, wenn $\mu \to -\infty$ (vgl. wieder Gleichung (5.10)). Folglich liegt jetzt das Maximum von $\overline{f}$ bei Beschränkung der Varianz auf $\sigma^2_{\min}$ bei $\mu = -\infty$. Abbildung 5.5 zeigt das Verhalten der Dichtefunktion $\overline{f}$ für $x = 0$ bzw. x nahe der Nullgrenze der gestutzten Normalverteilung.

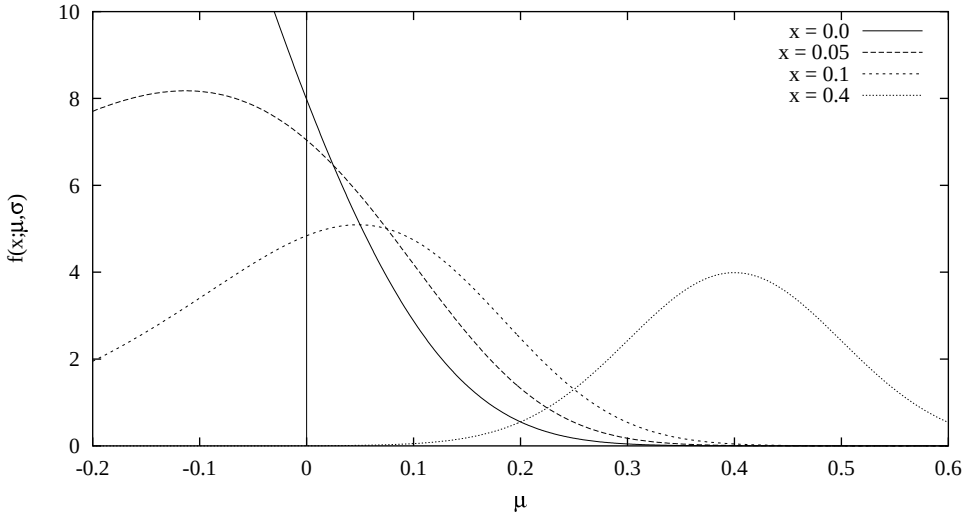


Abbildung 5.5: $\overline{f}(x; \mu, \sigma)$ in Abhängigkeit von μ für $\sigma = \sigma_{\min} = 0.1$ und verschiedene x

Das eigentliche Problem, das an dieser Stelle zutage tritt, liegt in der Tatsache, dass wir eine linksseitig bei 0 gestutzte Normalverteilung für Daten verwenden, die sehr häufig den Wert 0 annehmen, also auf dem Abschneidepunkt selbst liegen. Falls sogar alle Daten gleich 0 sind, beträgt auch der statistische Erwartungswert 0, während der analytische Erwartungswert von $\overline{f}$ nie null werden kann (vgl. Abschnitt 5.2.4).

Verschiebung des Abschneidepunktes

Die von uns vorgeschlagene Modifikation besteht nun darin, die Normalverteilung nicht exakt bei $x = 0$ zu stutzen, sondern etwas links davon bei $x = -\varepsilon$ ($\varepsilon > 0$). Damit liegen an der kritischen Grenze keine Daten mehr und der Fall $V = E^2 = 0$ kann genauso behandelt werden wie der unproblematischere Fall $V = E^2 > 0$. Denn bei einer Verschiebung des Abschneidepunktes um z. B. $\varepsilon = 0.1$ nach links entspricht die Dichte mit $x = 0.1$ in Abbildung 5.5 von der Form her einer Dichte mit $x = 0$, verschoben um ebenfalls 0.1 nach links auf der μ-Achse. Der Nachteil einer solchen Verteilung besteht lediglich darin, dass beim Generieren von Daten wieder negative Ausgaben entstehen können; bei kleinem ε können wir diesen Fehler jedoch vernachlässigen.

Die Angleichung der für die gestutzte Normalverteilung in den vorangegangen Abschnitten aufgestellten Gleichungen, die wir zum Reestimieren der Parameter benötigen, soll hier nur kurz skizziert werden, da sich qualitativ nichts Wesentliches an den Aussagen und Verfahren ändert.

Wir definieren die bei $-\varepsilon$ gestutzte Normaldichte als

$$\tilde{f}(x;\mu,\sigma) = \begin{cases} 0 & x < -\varepsilon \\ \dfrac{1}{\tilde{a}(\mu,\sigma)} f(x;\mu,\sigma) & x \geq -\varepsilon \end{cases}$$

mit

$$\tilde{a}(\mu,\sigma) = \int_{-\varepsilon}^{\infty} f(x;\mu,\sigma)\,dx \;=\; \Phi(\frac{\mu+\varepsilon}{\sigma})\,.$$

Analog zu den partiellen Ableitungen (5.12) und (5.13) ergibt sich für $\tilde{a}(\mu,\sigma)$

$$\begin{aligned} \tilde{a}'(\mu) &= f(-\varepsilon;\mu,\sigma)\,, \\ \tilde{a}'(\sigma) &= -\frac{\mu+\varepsilon}{\sigma} f(-\varepsilon;\mu,\sigma)\,, \end{aligned}$$

und damit lassen sich die Ableitungen $\tilde{f}'(\mu)$ und $\tilde{f}'(\sigma)$ bestimmen. Letztendlich gelangen wir zu folgenden abgewandelten Gleichungen (5.23) und (5.24):

$$\begin{aligned} &\bar{\sigma}^2 = V + \varepsilon E - \bar{\mu}(E+\varepsilon)\,, \\ &p(\bar{\mu}) := E - \bar{\mu} - (V + \varepsilon E - \bar{\mu}[E+\varepsilon])\,\tilde{f}(-\varepsilon;\bar{\mu},\sqrt{V+\varepsilon E-\bar{\mu}[E+\varepsilon]}) = 0 \end{aligned}$$

bzw. mit $\tilde{E} := E + \varepsilon$ und $\tilde{V} := V + \varepsilon E$

$$\begin{aligned} &\bar{\sigma}^2 = \tilde{V} - \bar{\mu}\tilde{E}\,, \\ &p(\bar{\mu}) = E - \bar{\mu} - (\tilde{V} - \bar{\mu}\tilde{E})\,\tilde{f}(-\varepsilon;\bar{\mu},\sqrt{\tilde{V}-\bar{\mu}\tilde{E}}) = 0\,. \end{aligned}$$

Als rechte Definitionsgrenze der so veränderten Funktion $p(\mu)$ erhalten wir damit $\tilde{V}/\tilde{E}$, während sich die numerische Nullstellensuche auf $\mu < E$ beschränken kann (vgl. (5.27) und entsprechende Bemerkungen). Die linke Grenze kann analog zu (5.29) berechnet werden, wobei für die veränderte Funktion $p(\mu)$ jetzt $\Phi(\frac{\mu+\varepsilon}{\sqrt{\tilde{V}-\mu\tilde{E}}})$ nicht 0 werden darf. Die numerische Nullstellensuche erfolgt schließlich zwischen den Grenzen

$$\begin{aligned} \mu_L &= C\sqrt{\tilde{V} + \varepsilon\tilde{E} + \frac{C^2\tilde{E}^2}{4}} - \frac{C^2\tilde{E}}{2} - \varepsilon\,, \\ \mu_R &= E\,. \end{aligned}$$

Mit Einsatz der Dichte $\tilde{f}$ erhöht sich auch die numerische Stabilität von $p(\mu)$ bei sehr kleinen Ausgabewerten O_t; die Nullstelle kann dann auch für große Werte $V \gg E^2$ ohne Probleme bestimmt werden. (vgl. Abschnitt 5.2.3).

5.2.6 Erzeugung von Zufallszahlen

Nachdem wir die Reestimierungsformeln zum Training eines HMM an die Dichtefunktion der gestutzten Normalverteilung angepasst haben, soll an dieser Stelle noch kurz auf das Erzeugen

von entsprechenden Zufallszahlen eingegangen werden. Solche Zufallszahlen müssen generiert werden, wenn wir ein HMM zum Erzeugen von Datensequenzen benutzen möchten.

Die klassische Methode zur Erstellung von $\mathcal{N}(\mu, \sigma^2)$-verteilten Zufallszahlen besteht darin, aus gleichverteilten Zufallszahlen U zunächst $\mathcal{N}(0,1)$-verteilte Zufallszahlen Y zu produzieren. Die Transformation $Z = \mu + \sigma Y$ führt dann zu den gewünschten Zahlen. [12] enthält eine ausführliche Beschreibung der zahlreichen, meist effizienten Algorithmen und Methoden für das Erzeugen von normalverteilten Zufallszahlen.

Die einfachste Möglichkeit, Zufallszahlen einer gestutzten Normalverteilung zu erzeugen, besteht darin, normalverteilte Zahlen zu generieren und diejenigen darunter zu ignorieren, die im abgeschnittenen Teil der Verteilung liegen. Dieses Vorgehen hat den Vorteil, dass auf die meist schnellen und erprobten Zufallszahlengeneratoren in Programm-Bibliotheken zurückgegriffen werden kann. Die Effizienz geht natürlich dann verloren, wenn der abgeschnittene Teil der Normalverteilung gegenüber dem Teil, in dem die Dichte > 0 ist, extrem groß ist, d. h. wenn μ weit im negativen Bereich liegt. Aus diesem Grund verwenden wir in unserem Programm die Methode der inversen Transformation, mit der auch ohne deutliche Laufzeiterhöhung Zufallszahlen in einem beschränkten Intervall erzeugt werden können. Bei der inversen Transformation (siehe [12], S. 149 ff.) wird die Tatsache genutzt, dass die Umkehrfunktion $F^{-1}(u)$ einer Verteilungsfunktion $F(z)$ angewandt auf eine $(0,1)$-gleichverteilte Zufallszahl U eine Zufallszahl mit der gewünschten Verteilung F ergibt. Ersetzen wir das Argument der Umkehrfunktion durch die auf das Intervall $[F(a), F(b)]$ transformierte Zufallszahl $U' = F(a) + [F(b) - F(a)]U$, führt dies zu einer Zufallszahl im Intervall $[a, b]$ mit Verteilung F. Die Umkehrfunktion der Normalverteilung kann mit Hilfe einer Approximation nach Hastings berechnet werden [12].

5.3 Alternative Dichtefunktionen

Nach den umfangreichen Darstellungen des Abschnitts 5.2 stellt sich die Frage, ob es nicht andere Verteilungen gibt, die die gewünschten Anforderungen erfüllen und nur positive Zufallszahlen abbilden, dabei aber einfacher in ihrer Verwendung sind. Im Rahmen dieser Arbeit wurden verschiedene Verteilungen als Alternativen zur gestutzten Normalverteilung in Erwägung gezogen und aus den folgenden Gründen nicht weiter betrachtet, wobei natürlich auch die Form einer Dichtefunktion eine große Rolle spielt (Einzelheiten zu den angesprochenen Verteilungen finden sich in z. B. in [16, 19, 35]):

- Alle Verteilungen, die nur einem Parameter unterliegen, erscheinen uns nicht flexibel genug in der Abbildung beliebiger Daten als Ausgaben eines HMM-Zustands. Bei diesen Verteilungen ist die Abhängigkeit von Mittelwert und Varianz zu groß.
- Der Einsatz von komplexeren Verteilungen scheitert daran, dass die Differentiation der entsprechenden Q-Funktionen nach den Parametern der Verteilungen zu analytisch noch schwierigeren Gleichungen führt (vgl. Anschnitt 5.1 und Gleichung (5.5)). Darunter fallen z. B. die Lognormalverteilung oder die Gamma-Verteilung und ihre spezielle Form, die χ^2-Verteilung, für die im Gegensatz zur Normalverteilung auch keine expliziten Maximum-Likelihood-Schätzer berechnet werden können.

- Aus der Transformation $|X|$ einer normalverteilten Zufallsvariablen X entsteht eine „geklappte“ Normalverteilung, die ähnliche Eigenschaften hat wie die linksseitig bei 0 gestutzte Normalverteilung. Die Dichtefunktion von $|X|$ berechnet sich aus

$$\tilde{f}(x) = \begin{cases} 0 & x < 0 \\ f(-x) + f(x) & x \geq 0 \end{cases},$$

wobei f wieder die in Abschnitt 5.2.1 definierte Dichte der Normalverteilung bezeichnet. Gegenüber der Dichte $\bar{f}$ der gestutzten Verteilung, in der die nur numerisch lösbare Funktion Φ steckt, ist $\tilde{f}$ analytisch zu berechnen, und ebenso deren Ableitungen nach den Parametern μ und σ. Die Differentiation der entsprechenden Q-Funktion führt jedoch wieder auf eine implizite Nullstellengleichung, in der das gesuchte μ sogar nichtlinear mit jeder Ausgabe O_t verknüpft ist, so dass ein numerisches Lösungsverfahren in jeder Iteration über alle Ausgaben O_t laufen muss. Die damit verbundene Laufzeiterhöhung eines solchen Verfahrens schließt die weitere Verwendung dieser Dichte praktisch aus (vgl. Abschnitt 5.2.3, vorletzter Absatz).

Kapitel 6

Erweitertes HMM mit Ausgabe-Klassen

In Abschnitt 3.3.2 hatten wir bereits festgestellt, dass bei den vollständigen Spargeld-Sequenzen, denen ausschließlich zugeteilte Bausparverträge zugrunde liegen und die wir zur Clusterung und zum Trainieren der Modelle einsetzen wollen, die Summe der Einträge einer Sequenz praktisch nie unter einem festen Wert liegt. Dies resultiert aus den Voraussetzungen der Zuteilung, bei der ein Vertrag eine vorgegebene Mindestsparleistung erbracht haben muss, wobei sich die Sparleistung aus den eingezahlten Beträgen und den Zinsen zusammensetzt (vgl. Abschnitt 3.1). Andererseits wird ein Vertrag, bei dem die nötigen Sparleistungen erbracht wurden, mit großer Wahrscheinlichkeit die Sparphase beenden. Ein Sparer macht i. d. R. sein Verhalten vor allem zum Ende der Sparphase hin davon abhängig, wieviel er bereits angespart hat bzw. wieviel er noch einzahlen muss, um den Anspruch an das Darlehen zu einem bestimmten Zeitpunkt zu erreichen. Ein Bausparer, der bereits kurz vor der Zuteilung steht, wird deshalb z. B. nur noch kleine Summen einzahlen. In dem HMM, das eine solche Sequenz repräsentiert, müsste die Wahrscheinlichkeit, dass die darunterliegende Zustandsfolge zu diesem Zeitpunkt in den Endzustand wechselt, größer sein als bei einer Sequenz, die bis dahin noch wenig angespart hat.

Sowohl in einem HMM, wie wir es in Kapitel 2 vorgestellt haben, als auch in den in Abschnitt 2.6 angesprochenen allgemeineren Hidden-Markov-Modellen ist eine solche Abhängigkeit des stochastischen Prozesses von der Summe der bisherigen Ausgaben höchstens implizit durch das Training des Modells mit den entsprechenden Sequenzen gegeben. Wir stellen deshalb in diesem Kapitel ein erweitertes HMM vor, bei dem die Summen der Sequenzeinträge explizit berücksichtigt werden können. Dazu definieren wir aber zunächst unabhängig von unserer speziellen Anwendung eine neue Modellklasse und stellen für diese die entsprechend angepassten Basisalgorithmen bereit.

In Kapitel 7 werden wir neben den klassischen Hidden-Markov-Modellen die erweiterten Modelle zur Abbildung der Spargeld-Sequenzen verwenden und beide Modellklassen in Bezug auf unsere Anwendung vergleichen.

6.1 Modellbeschreibung und Definitionen

Die Grundidee der Modellerweiterung besteht darin, dass der Wechsel von einem Zustand in einen anderen nicht mehr durch eine feste Übergangswahrscheinlichkeit beschrieben wird, sondern darüberhinaus von der Summe der Einträge der bis dahin ausgegebenen Teilsequenz abhängen soll. Diese Abhängigkeit wird mit Hilfe einer Klasseneinteilung aller möglichen Ausgabesummen realisiert; die im Folgenden beschriebene Modellerweiterung gilt jedoch ganz allgemein für jede Abbildung, die eine (Teil-)Sequenz auf einen Index einer Klasse aus einer festen und endlichen Menge von Klassen abbildet:

Definition 6.1 *$\mathcal{U} = U_1, U_2, \ldots, U_L$ bezeichne eine feste Menge von Klassen, und die Abbildung $cl : \mathbb{R}^t \mapsto \mathbb{N}$ weise jeder Teilsequenz $O_{(t)} = O_1 \cdots O_t$, $t \in \{1, 2, \ldots, T\}$, den Index einer Klasse aus $\mathcal{U}$ zu. Die Klassen in $\mathcal{U}$ nennen wir Ausgabe-Klassen und bezeichnen mit $p_t = cl(O_{(t)}) \in \{1, \ldots, L\}$ die Ausgabe-Klasse der Teilsequenz $O_{(t)}$.*

Damit ist mit jeder Sequenz $O_1 \cdots O_T$ bei gegebener Klasseneinteilung auch eine eindeutige Folge von Ausgabe-Klassen $p_1 \cdots p_T$ festgelegt. Wir werden im Weiteren stets davon ausgehen, dass $cl(O_{(t)})$ in konstanter Zeit ausgewertet werden kann, insbesondere also nicht von der Anzahl L der Ausgabe-Klassen abhängt.

Wir definieren nun das erweiterte Modell folgendermaßen, wobei wir uns auf die in Abschnitt 2.1 eingeführten Notationen für Hidden-Markov-Modelle beziehen:

Definition 6.2 (Erweitertes HMM mit Ausgabe-Klassen) *Wir ersetzen in einem HMM jede Übergangswahrscheinlichkeit a_{ij} der versteckten Markov-Kette von Zustand S_i nach Zustand S_j durch den L-dimensionalen Vektor*

$$a_{ij} := (a_{i1j}, a_{i1j}, \ldots, a_{iLj}) .$$

Der l-te Eintrag bezeichne die bedingte Wahrscheinlichkeit, dass der stochastische Prozess zum Zeitpunkt t in den Zustand S_j wechselt, gegeben den Zustand S_i, das Modell λ und die Ausgabe-Klasse l der bisher vom Modell ausgegebenen Teilsequenz $O_{(t)}$:

$$a_{ilj} = P(q_{t+1} = j | p_t = l, q_t = i, \lambda) . \tag{6.1}$$

Dabei gelte für jedes $i \in \{1, \ldots, N\}$ und $l \in \{1, \ldots, L\}$ die stochastische Nebenbedingung $\sum_{j=1}^{N} a_{ilj} = 1$. Das in dieser Form modifizierte HMM bezeichnen wir als ein erweitertes HMM mit (L) Ausgabe-Klassen (AK-HMM).

Damit vergrößert sich der Parameterraum eines HMM um $(L-1)N^2$ zusätzliche Parameter, denn aus der ursprünglichen Matrix der Wahrscheinlichkeitsverteilung der Zustandsübergänge wird im erweiterten Modell die dreidimensionale $N \times L \times N$-Matrix $A = \{a_{ilj}\}$. Für alle anderen Modellparameter verwenden wir die gewohnten Notationen (vgl. Abschnitt 2.1).

Im Übergangsgraphen eines in dieser Form definierten HMM verlaufen zwischen jedem gerichtet verbundenen Knotenpaar jeweils L Kanten mit i. d. R. unterschiedlichen Gewichten, die den jeweiligen bedingten Wahrscheinlichkeiten entsprechen (vgl. Abbildung 6.1).

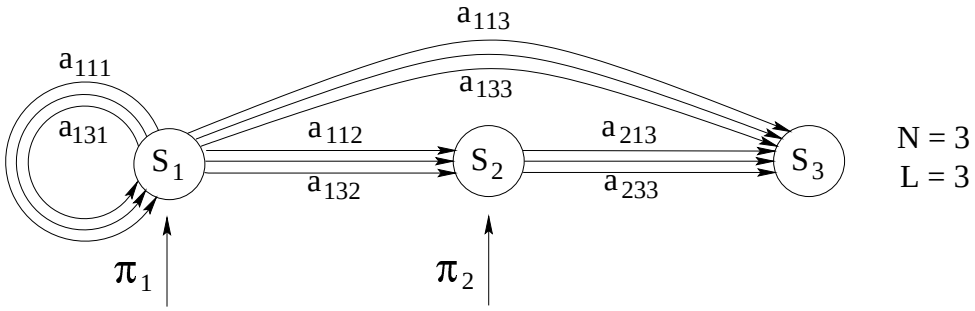


Abbildung 6.1: Graph eines erweiterten HMM mit $L = 3$ bedingten Übergangsklassen

Das beschriebene Modell besitzt die von uns eingangs gewünschte Abhängigkeit der Übergangswahrscheinlichkeiten von der Summe der bisher ausgegebenen Daten, wenn wir für $cl(O_{(t)})$ eine Abbildung der Summe $\sum_{\tau=1}^{t} O_\tau$ auf eine Intervalleinteilung in $\mathbb{R}$ wählen.

Es stellt sich zunächst aber die Frage, ob mit dem erweiterten HMM genauso gearbeitet werden kann wie mit dem klassischen Modell, denn schließlich hatten wir bei der Entwicklung der relevanten Algorithmen an vielen Stellen implizit die in Abschnitt 2.1 getroffene Markov-Annahme – d. h. die Wahrscheinlichkeit, von einem Zustand zum nächsten zu wechseln, hängt nur von diesen beiden Zuständen ab – benutzt. Gerade diese gilt aber für das erweiterte Modell offensichtlich nicht mehr. Aus diesem Grund sollen in den folgenden Abschnitten die Basisalgorithmen ausführlich geprüft und, soweit es möglich ist, so angepasst werden, dass sie die gewünschten Berechnungen liefern, ohne die Komplexität deutlich zu erhöhen.

6.2 Forward-Backward-Algorithmus

Analog zum klassischen Hidden-Markov-Modell bezeichnen wir mit der Forward-Variablen $\alpha_t(i)$ die Wahrscheinlichkeit (bzw. Likelihood im Fall von stetigen Ausgaben), dass die Teilsequenz $O_1 O_2 \cdots O_t$ ausgegeben wird und sich die Zustandsfolge zum Zeitpunkt t in Zustand S_i befindet, gegeben die Modellparameter λ:

$$\alpha_t(i) = P(O_1 O_2 \cdots O_t, q_t = i | \lambda) .$$

Die Berechnung dieser Variablen für alle Zustände i und Zeitschritte t erfolgt wiederum rekursiv:

1. $t = 1$:

$$\alpha_1(i) = P(O_1, q_1 = i | \lambda) = \pi_i b_i(O_1), \qquad 1 \leq i \leq N .$$

2. $t < T$:

$$
\begin{aligned}
\alpha_{t+1}(i) &= P(O_1 \cdots O_{t+1}, q_{t+1} = i|\lambda) \\
&= P(O_1 \cdots O_t, q_{t+1} = i|\lambda) P(O_{t+1}|O_1 \cdots O_t, q_{t+1} = i, \lambda) \\
&= P(O_1 \cdots O_t, q_{t+1} = i|\lambda)\, b_i(O_{t+1}) \\
&= \left[\sum_{j=1}^{N} P(O_1 \cdots O_t, q_t = j, q_{t+1} = i|\lambda)\right] b_i(O_{t+1}) \\
&= \left[\sum_{j=1}^{N} P(O_1 \cdots O_t, q_t = j|\lambda) P(q_{t+1} = i|O_1 \cdots O_t, q_t = j, \lambda)\right] b_i(O_{t+1}) \\
&= \left[\sum_{j=1}^{N} \alpha_t(j) P(q_{t+1} = i|O_1 \cdots O_t, q_t = j, \lambda)\right] b_i(O_{t+1})\,.
\end{aligned}
$$

Beim klassischen HMM gilt $P(q_{t+1} = i|O_1 \cdots O_t, q_t = j, \lambda) = P(q_{t+1} = i|q_t = j, \lambda) = a_{ji}$, da der Wechsel von Zustand S_j nach Zustand S_i unabhängig ist von der bereits erzeugten Teilsequenz $O_1 \cdots O_t$. In dem erweiterten Modell ist jedoch die Wahrscheinlichkeit, im Zeitschritt t von S_i nach S_j zu wechseln, eine Funktion von $p_t = cl(O_{(t)})$. Da aber mit der Teilsequenz auch die entsprechende Ausgabe-Klasse l bekannt ist, gilt:

$$P(q_{t+1} = i|O_1 \cdots O_t, q_t = j, \lambda) = P(q_{t+1} = i|p_t = l, q_t = j, \lambda) = a_{jli}\,.$$

Zusammenfassend berechnet sich $\alpha_{t+1}(i)$ also wie folgt:

$$
\begin{aligned}
l &= cl(O_{(t)})\,, \\
\alpha_{t+1}(i) &= \left[\sum_{j=1}^{N} \alpha_t(j) a_{jli}\right] b_i(O_{t+1})\,, \qquad 1 \le t \le T-1,\ 1 \le i \le N\,.
\end{aligned}
$$

Die Berechnung der $\alpha_{t+1}(i)$ wird auf gewohnte Weise in jedem Zeitschritt für alle Zustände i durchgeführt. Als letzter Schritt folgt dann die Berechnung der Wahrscheinlichkeit der Sequenz, gegeben λ:

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i)\,.$$

Abbildung 6.2 verdeutlicht die Vorgehensweise der Bestimmung von $\alpha_{t+1}(i)$ aus den Größen aller möglicher Vorgängerknoten S_j. Dabei entscheidet die pro Zeitschritt t einmal bestimmte Ausgabe-Klasse l darüber, welche der Übergangswahrscheinlichkeiten in die Variable einfließen, in der Abbildung dargestellt durch durchgezogene Pfeile.

Im Vergleich zum klassischen HMM bleibt die Komplexität der so veränderten Prozedur zur Bestimmung der Forward-Variablen erhalten ($O(N^2T)$), da $l = cl(O_{(t)})$ nach Voraussetzung in jedem Schritt in konstanter Zeit berechnet werden kann.

Der Algorithmus zur Berechnung der Backward-Variablen lässt sich leider nicht auf die gleiche und einfache Art übertragen, wie wir im Folgenden sehen werden. Sei auch hier wieder $\beta_t(i)$ die

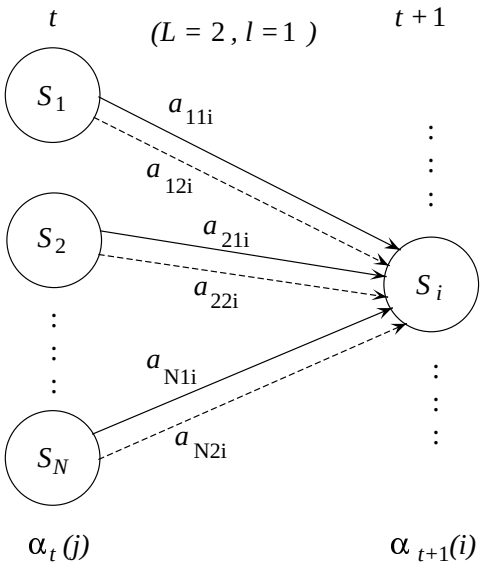


Abbildung 6.2: Rekursive Berechnung der $\alpha_{t+1}(i)$

Wahrscheinlichkeit, dass die Teilsequenz $O_{t+1} \cdots O_T$ ausgegeben wird, gegeben Zustand S_i zum Zeitpunkt t und die Modellparameter λ:

$$\beta_t(i) = P(O_{t+1} \cdots O_T | q_t = i, \lambda) \,.$$

Beim Versuch, die Berechnung der $\beta_t(i)$ auf die übliche Weise auf alle $\beta_{t+1}(j)$ zurückzuführen, gelangen wir zunächst zu folgender Gleichung:

$$\begin{aligned} \beta_t(i) &= P(O_{t+1} \cdots O_T | q_t = i, \lambda) \\ &= \sum_{j=1}^{N} P(O_{t+1} \cdots O_T, q_{t+1} = j | q_t = i, \lambda) \\ &= \sum_{j=1}^{N} P(q_{t+1} = j | q_t = i, \lambda) P(O_{t+1} \cdots O_T | q_{t+1} = j, q_t = i, \lambda) \end{aligned}$$

Das Produkt der beiden Wahrscheinlichkeiten in der letzten Zeile entspricht beim klassischen HMM gerade dem Ausdruck $a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)$. Für das erweiterte Modell dagegen können weder $P(q_{t+1} = j | q_t = i, \lambda)$ direkt berechnet noch $P(O_{t+1} \cdots O_T | q_{t+1} = j, q_t = i, \lambda)$ als Produkt von $b_j(O_{t+1}) \beta_{t+1}(j)$ formuliert werden, da zum einen eine von der Ausgabe-Klasse unabhängige Übergangswahrscheinlichkeit nicht vorliegt und zum anderen die Ereignisse, die den Wahrscheinlichkeiten $\beta_{t+1}(j)$ und $b_j(O_{t+1})$ zugrunde liegen, stochastisch nicht unabhängig sind.

Um die oben genannten Probleme zu umgehen, definieren wir eine modifizierte Variable β'_t als die Wahrscheinlichkeit, dass die Teilsequenz $O_{t+1} \cdots O_T$ ausgegeben wird, gegeben Zustand S_i zum Zeitpunkt t, die Teilsequenz $O_{(t)} = O_1 \cdots O_t$ und die Modellparameter λ:

$$\beta'_t(i) = P(O_{t+1} \cdots O_T | q_t = i, O_1 \cdots O_t, \lambda) \,.$$

Diese Variable kann jetzt rekursiv berechnet werden, denn es gilt:

$$\begin{aligned}
\beta'_t(i) &= P(O_{t+1} \cdots O_T | q_t = i, O_{(t)}, \lambda) \\
&= \sum_{j=1}^{N} P(O_{t+1} \cdots O_T, q_{t+1} = j | q_t = i, O_{(t)}, \lambda) \qquad (6.2a) \\
&= \sum_{j=1}^{N} P(q_{t+1} = j | q_t = i, O_{(t)}, \lambda) P(O_{t+1} \cdots O_T | q_t = i, q_{t+1} = j, O_{(t)}, \lambda) \\
&= \sum_{j=1}^{N} a_{ip_{tj}} P(O_{t+1} \cdots O_T | q_t = i, q_{t+1} = j, O_{(t)}, \lambda) \\
&= \sum_{j=1}^{N} a_{ip_{tj}} P(O_{t+1} | q_t = i, q_{t+1} = j, O_{(t)}, \lambda) P(O_{t+2} \cdots O_T | O_{t+1}, q_t = i, q_{t+1} = j, O_{(t)}, \lambda) \\
&= \sum_{j=1}^{N} a_{ip_{tj}} b_j(O_{t+1}) P(O_{t+2} \cdots O_T | q_{t+1} = j, O_{(t+1)}, \lambda) \\
&= \sum_{j=1}^{N} a_{ip_{tj}} b_j(O_{t+1}) \beta'_{t+1}(j) \,. \qquad (6.2b)
\end{aligned}$$

Analog zur Berechnung der β_t im ursprünglichen Modell erhalten wir damit folgende Rekursion zur Bestimmung aller β'_t:

1. $t = T$:

$$\beta'_T(i) = 1, \qquad 1 \leq i \leq N \,.$$

2. $t < T$:

$$\begin{aligned}
l &= p_t = cl(O_{(t)}) \,, \\
\beta'_t(i) &= \sum_{j=1}^{N} a_{ilj} b_j(O_{t+1}) \beta'_{t+1}(j) \,, \qquad t = T-1, \ldots, 1, \; 1 \leq i \leq N \,.
\end{aligned}$$

Im nächsten Abschnitt werden wir sehen, dass mit Hilfe der modifizierten Backward-Variablen β'_t ganz analog zum klassischen HMM die Größen berechnet werden können, die wir für den Baum-Welch-Algorithmus benötigen.

6.3 Baum-Welch-Algorithmus

Eine zentrale Rolle beim Trainieren der Parameter eines Modells mittels des Baum-Welch-Algorithmus spielen die Größen $\gamma_t(i)$ und $\xi_t(i,j)$, die jeweils aus den Forward- und Backward-Variablen berechnet werden können. Diese Berechnungen sind auch für das erweiterte Modell mit

den modifizierten Variablen $\alpha_t(i)$ und β'_t möglich. Dazu betrachten wir zunächst deren Produkt:

$$\begin{aligned}\alpha_t(i)\beta'_t(i) &= P(O_1 \cdots O_t, q_t = i|\lambda)\, P(O_{t+1} \cdots O_T|q_t = i, O_1 \cdots O_t, \lambda) \\ &= P(O_1 \cdots O_t, O_{t+1} \cdots O_T, q_t = i|\lambda) \\ &= P(O, q_t = i|\lambda)\,. \qquad (6.3)\end{aligned}$$

(6.4)

Dies ist aber gerade wieder die Modellwahrscheinlichkeit, die vollständige Sequenz auszugeben und zur Zeit t im Zustand S_i zu sein, und somit liefert die Summierung dieser Produkte über alle Zustände wieder die Wahrscheinlichkeit der vollständigen Sequenz, gegeben λ:

$$\sum_{i=1}^{N} \alpha_t(i)\beta'_t(i) = P(O|\lambda)\,, \qquad t \in \{1, \ldots T\}\,. \qquad (6.5)$$

An dieser Stelle wird deutlich, dass wir mit den Forward- und Backward-Variablen des erweiterten Modells alle Größen und Formeln für den Baum-Welch-Algorithmus anpassen können.

Bezeichne $\gamma_t(i)$ wieder die Wahrscheinlichkeit, zum Zeitpunkt t im Zustand i zu sein, gegeben die Sequenz O und die Modellparameter. Dann folgt aus den Gleichungen (6.4) und (6.5):

$$\gamma_t(i) = P(q_t = i|O, \lambda) = \frac{\alpha_t(i)\beta'_t(i)}{\sum_{i=1}^{N} \alpha_t(i)\beta'_t(i)}\,. \qquad (6.6)$$

Auch die Berechnung der Größe $\xi_t(i,j)$, der Wahrscheinlichkeit eines Zustandswechsels von S_i nach S_j zum Zeitpunkt t, gegeben O und λ, erfolgt in bekannter Weise, denn es gilt

$$\begin{aligned}P(q_t = i, q_{t+1} = j|O, \lambda)\, P(O|\lambda) &= P(O, q_t = i, q_{t+1} = j|\lambda) \\ &= P(O_{(t)}, q_t = i|\lambda)\, P(O_{t+1} \cdots O_T, q_{t+1} = j|O_{(t)}, q_t = i, \lambda) \\ &= \alpha_t(i) a_{ip_ij} b_j(O_{t+1}) \beta'_{t+1}(j)\,,\end{aligned}$$

wobei die letzte Umformung bereits in Abschnitt 6.2 mit der Gleichheit von (6.2a) und (6.2b) gezeigt wurde. Damit ergibt sich

$$\xi_t(i,j) = P(q_t = i, q_{t+1} = j|O, \lambda) = \frac{\alpha_t(i) a_{ip_ij} b_j(O_{t+1}) \beta'_{t+1}(j)}{P(O|\lambda)}\,, \qquad (6.7)$$

und es gilt

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j)\,. \qquad (6.8)$$

Die Summen der $\gamma_t(i)$ und $\xi_t(i,j)$ jeweils über t können wieder als Erwartungswerte interpretiert werden

$$\begin{aligned}\sum_{t=1}^{T-1} \gamma_t(i) &= \text{erwartete Anzahl der Übergänge aus } S_i \\ \sum_{t=1}^{T-1} \xi_t(i,j) &= \text{erwartete Anzahl der Übergänge von } S_i \text{ nach } S_j\end{aligned}$$

Mit Hilfe der obigen Formeln und Größen ergeben sich folgende Reestimierungsformeln als natürliche Schätzer $\bar{\lambda}$ der Parameter eines erweiterten HMM mit diskreten Ausgaben bei gegebenen Parametern λ:

$$\begin{aligned}
\bar{\pi}_i &= \text{erwartete Wahrscheinlichkeit, zur Zeit } t = 1 \text{ in } S_i \text{ zu sein} \\
&= \gamma_1(i)\,, && (6.9a) \\
\bar{a}_{ilj} &= \frac{\text{erwartete Anzahl der Übergänge } S_i \text{ nach } S_j \text{ in Ausgabe-Klasse } l}{\text{erwartete Anzahl der Übergänge aus } S_i \text{ in Ausgabe-Klasse } l} \\
&= \frac{\sum\limits_{\substack{t=1 \\ p_t=l}}^{T-1} \xi_t(i,j)}{\sum\limits_{\substack{t=1 \\ p_t=l}}^{T-1} \gamma_t(i)}\,, && (6.9b) \\
\bar{b}_{jm} &= \frac{\text{erwartete Anzahl von Ausgaben } v_m \text{ in } S_j}{\text{erwartete Anzahl, in } S_j \text{ zu sein}} \\
&= \frac{\sum\limits_{\substack{t=1 \\ O_t=v_m}}^{T} \gamma_t(j)}{\sum\limits_{t=1}^{T} \gamma_t(j)}\,. && (6.9c)
\end{aligned}$$

Dabei sind die Gleichungen für $\bar{\pi}_i$ und $\bar{b}_{jm}$ – abgesehen von der veränderten Berechnung der $\gamma_t(i)$ – identisch mit denen für das klassische HMM, während zum Schätzen der a_{ilj} nur die Übergänge gezählt werden, für die im Zustand i die Ausgabe-Klasse l angenommen wird. Die Gleichungen der Ausgabeparameter eines HMM mit stetigen Ausgaben können ganz analog erstellt werden: In (2.25c) bis (2.25e) werden einfach die entsprechend mit den erweiterten Forward-Backward-Variablen $\alpha_t(i)$ und $\beta'_t(i)$ formulierten $\gamma_t(i)$ und $\zeta_t(j,m)$ eingesetzt.

Die Komplexität zur Berechnung der dreidimensionalen Matrix A beträgt $O(TN^2) + O(LN^2)$, da in einem ersten Schritt alle Zähler und Nenner unabhängig von L berechnet und danach die LN^2 Einträge von A jeweils in konstanter Zeit zugewiesen werden können. Unter der realistischen Annahme, dass die Anzahl der Ausgabe-Klassen kleiner ist als die Sequenzlänge, ändert sich damit die asymptotische Laufzeit des Baum-Welch-Algorithmus gegenüber der ursprünglichen Version nicht (vgl. Abschnitt 4.2.2). Der Speicherbedarf für die Übergangswahrscheinlichkeiten steigt jedoch um den Faktor L.

Die intuitiv hergeleiteten Reestimierungsformeln erfüllen wieder automatisch die stochastischen Nebenbedingungen für die Parameter:

$$\sum_{j=1}^{N} \bar{\pi}_j = \sum_{j=1}^{N} \bar{a}_{ilj} = \sum_{m=1}^{M} \bar{b}_{im} = 1 \quad i \in \{1,\ldots,N\}\,,\ l \in \{1,\ldots,L\}\,.$$

Schließlich sind alle Formeln und Algorithmen des erweiterten Modells konsistent mit denjenigen des ursprünglichen Modells: Wenn wir die Anzahl der Ausgabe-Klassen L auf 1 setzen, haben wir in allen Gleichungen exakte Übereinstimmung mit dem ursprünglichen HMM .

Zusammenfassend können wir feststellen, dass für das erweiterte Modell die Basisalgorithmen zur Berechnung und Maximierung der Wahrscheinlichkeit $P(O|\lambda)$ bzw. der Likelihood $L(O|\lambda)$ auf einfache Art und ohne die Komplexitätsklasse zu ändern angepasst werden können. Natürlich ist damit noch nicht die Frage geklärt, ob das Reestimierungsverfahren in dieser Form noch konvergiert bzw. zu einem lokalen Maximum führt. Es stellt sich jedoch heraus, dass der Beweis des Baum-Welch-Algorithmus leicht auf das um Ausgabe-Klassen erweiterte HMM übertragbar ist, wie wir im nächsten Abschnitt sehen werden.

6.4 Konvergenz des erweiterten Baum-Welch-Algorithmus

Wir werden im Folgenden zeigen, dass auch die für das erweiterte Modell intuitiv hergeleiteten Reestimierungsformeln (6.9a) – (6.9c) die Likelihood mit jedem Schritt verbessern, bis ein kritischer Punkt oder das globale Maximum erreicht wird. Dazu folgen wir der Beweisskizze in Abschnitt 2.2.4 und passen diese an den entscheidenden Stellen an. Es wird leicht zu sehen sein, dass diese Anpassungen in der gleichen Form auf den entsprechenden Beweis für Hidden-Markov-Modelle mit stetigen Ausgaben, wie er z. B. in [20] zu finden ist, übertragen werden können.

Wir definieren für zwei Modelle λ und $\bar{\lambda}$ mit Ausgabe-Klassen wieder die Hilfsfunktion $\mathcal{Q}(\lambda, \bar{\lambda})$:

$$\mathcal{Q}(\lambda, \bar{\lambda}) = \sum_{s=1}^{S} P(Q_s, O|\lambda) \cdot \ln P(Q_s, O|\bar{\lambda}) \,, \tag{6.10}$$

wobei S wieder die Anzahl aller möglichen Zustandspfade der Länge T durch die Modelle und Q_s den s-ten Zustandspfad bezeichnet. Dann folgt mit Satz 2.3 und analog zur Beweisskizze in Abschnitt 2.2.4, dass eine Maximierung der Q-Funktion bezüglich $\bar{\lambda}$ die gewünschte Verbesserung der Likelihood erzielt.

Zur Berechnung der Q-Funktion muss jedoch geklärt werden, ob sich $P(Q_s, O|\lambda)$ wieder wie beim klassischen HMM als Produkt der entsprechenden Modellparameter berechnen lässt, da die Übergänge und die Ausgaben des erweiterten Modells im Allgemeinen ja nicht unabhängig sind. Seien $Q = q_1 q_2 \cdots q_T$ eine gegebene Zustandsfolge und $Q_{(t)} = q_1 q_2 \cdots q_t$ die entsprechende Teilfolge, dann gilt

$$\begin{aligned} P(Q, O|\lambda) &= P(Q_{(T-1)}, O_{(T-1)}|\lambda)\, P(q_T, O_T|Q_{(T-1)}, O_{(T-1)}, \lambda) \\ &= P(Q_{(T-1)}, O_{(T-1)}|\lambda)\, P(q_T|Q_{(T-1)}, O_{(T-1)}, \lambda)\, P(O_T|Q, O_{(T-1)}, \lambda) \\ &= P(Q_{(T-1)}, O_{(T-1)}|\lambda)\, a_{q_{T-1}p_{T-1}q_T} b_{q_T}(O_T) \\ &= \cdots \\ &= \pi_{q_1} b_{q_1}(O_1) \prod_{t=1}^{T-1} a_{q_t p_t q_{t+1}} b_{q_{t+1}}(O_{t+1}) \,, \end{aligned} \tag{6.11}$$

wobei p_t wieder die Ausgabe-Klasse der Sequenz O im Zeitpunkt t festlegt. Daraus folgt

$$\ln P(Q = Q_s, O|\overline{\lambda}) = \ln \overline{\pi}_{q_1} + \sum_{t=1}^{T-1} \ln \overline{a}_{q_t p_t q_{t+1}} + \sum_{t=1}^{T} \ln \overline{b}_{q_t}(O_t)\,, \tag{6.12}$$

und durch Einsetzen von (6.12) in (6.10) erhalten wir

$$\begin{aligned}\mathcal{Q}(\lambda, \overline{\lambda}) &= \sum_{Q=Q_1}^{Q_S} P(Q, O|\lambda) \ln \overline{\pi}_{q_1} + \sum_{Q=Q_1}^{Q_S} P(Q, O|\lambda) \sum_{t=1}^{T-1} \ln \overline{a}_{q_t p_t q_{t+1}} \\ &\quad + \sum_{Q=Q_1}^{Q_S} P(Q, O|\lambda) \sum_{t=1}^{T} \ln \overline{b}_{q_t}(O_t) \\ &=: \mathcal{Q}_\pi + \mathcal{Q}_a + \mathcal{Q}_b\,. \end{aligned} \tag{6.13}$$

Die so definierten Teilfunktionen $\mathcal{Q}_\pi$ und $\mathcal{Q}_b$ gleichen damit den entsprechenden Teilfunktionen in (2.19) und können damit wie in (2.20a) und (2.20c) umgeformt werden, während für $\mathcal{Q}_a$ gilt:

$$\begin{aligned}\mathcal{Q}_a &= \sum_{Q=Q_1}^{Q_S} P(Q, O|\lambda) \sum_{t=1}^{T-1} \ln \overline{a}_{q_t p_t q_{t+1}} \\ &= \sum_{i=1}^{N} \sum_{l=1}^{L} \sum_{j=1}^{N} \sum_{t=1}^{T-1} P(q_t = i, p_t = l, q_{t+1} = j, O|\lambda) \ln \overline{a}_{ilj} \\ &= P(O|\lambda) \sum_{i=1}^{N} \sum_{l=1}^{L} \sum_{j=1}^{N} \sum_{\substack{t=1 \\ p_t = l}}^{T-1} \xi_t(i,j) \ln \overline{a}_{ilj}\,, \end{aligned} \tag{6.14}$$

wobei die Größen $\gamma_t(i)$ und $\xi_t(i,j)$ jetzt nach (6.6) und (6.7) Funktionen der erweiterten Forward- und Backward-Variablen sind.

Damit ist die Q-Funktion wieder eine Summe unabhängiger Funktionen des Typs $F(x)$ aus Lemma 2.2 und wird genau dann maximiert, wenn die Parameter $\overline{\pi}_i$ und $\overline{b}_{jm}$ nach (2.21a) und (2.21c) und die bedingten Übergangswahrscheinlichkeiten $\overline{a}_{ilj}$ nach folgender Formel berechnet werden:

$$\overline{a}_{ilj} = \frac{\sum\limits_{\substack{t=1 \\ p_t = l}}^{T-1} \xi_t(i,j)}{\sum\limits_{k=1}^{N} \sum\limits_{\substack{t=1 \\ p_t = l}}^{T-1} \xi_t(i,k)} = \frac{\sum\limits_{\substack{t=1 \\ p_t = l}}^{T-1} \xi_t(i,j)}{\sum\limits_{\substack{t=1 \\ p_t = l}}^{T-1} \gamma_t(i)}\,. \tag{6.15}$$

Diese entsprechen wiederum den Reestimierungsformeln (6.9a) – (6.9c) im vorigen Abschnitt.

In Gleichung (6.13) wird deutlich, dass der Schlüssel zur Maximierung der Q-Funktion in deren Zerlegbarkeit besteht. Für Hidden-Markov-Modelle mit stetigen Ausgaben gilt diese Zerlegbarkeit genauso, und die hier gezeigte Beweisskizze ist deshalb direkt auf diesen Fall übertragbar. Dabei entsteht die gleiche Teilfunktion $\mathcal{Q}_a$, und somit gilt auch für das erweiterte HMM bei stetigen Ausgaben die Reestimierungsformel (6.15) für die $\overline{a}_{ilj}$. Bei den Gleichungen für die übrigen Parameter müssen nur die entsprechend anders berechneten $\gamma_t(i)$ und $\xi_t(i,j)$ eingesetzt werden (vgl. [17, 20]).

6.5 Viterbi-Algorithmus

In Abschnitt 2.2.2 wurde der Viterbi-Pfad definiert als die Zustandsfolge, die für eine gegebene Sequenz O bei gegebenem Modell λ am wahrscheinlichsten ist, also $P(Q, O|\lambda)$ maximiert. Da mit gegebener Sequenz O bei dem erweiterten HMM auch die Folge von Ausgabe-Klassen bekannt ist und nachdem die Forward-Variablen und Reestimierungsformeln erfolgreich angepasst werden konnten, können wir problemlos einen erweiterten Viterbi-Algorithmus aufstellen.

Für eine Zustandsfolge $Q = q_1 q_2 \cdots q_T$, die Sequenz O und das Modell λ definieren wir wieder die Größe

$$\delta_t(i) = \max_{q_1,q_2,\ldots,q_t} P(q_1 q_2 \cdots q_t = i, O_1 O_2 \cdots O_t|\lambda) .$$

Per Induktion und mit (6.11) gilt:

$$\delta_{t+1}(j) = [\max_i \delta_t(i) a_{ip_{ij}}] \cdot b_j(O_{t+1}) .$$

Damit kann der Viterbi-Algorithmus analog zum Vorgehen in Abschnitt 2.2.2 formuliert werden:

1. Initialisierung ($t = 1$):

$$\begin{aligned} \delta_1(i) &= \pi_i b_i(O_1) \qquad 1 \leq i \leq N , \\ \psi_1(i) &= 0 . \end{aligned}$$

2. Rekursion ($2 \leq t \leq T$):

$$\begin{aligned} l &= cl(O_{(t)}) , \\ \delta_t(j) &= \max_{1\leq i\leq N} [\delta_{t-1}(i) a_{ilj}] b_j(O_t) , \qquad 1 \leq j \leq N , \\ \psi_t(j) &= \operatorname*{argmax}_{1\leq i\leq N} [\delta_{t-1}(i) a_{ilj}] , \qquad 1 \leq j \leq N . \end{aligned}$$

3. Ende:

$$\begin{aligned} P^* &= \max_{1\leq i\leq N} [\delta_T(i)] , \\ q_T^* &= \operatorname*{argmax}_{1\leq i\leq N} [\delta_T(i)] . \end{aligned}$$

4. Backtracking:

$$q_t^* = \psi_{t+1}(q_{t+1}^*) , \qquad t = T-1, T-2, \ldots , 1 .$$

Auch hier bleibt die Komplexität des ursprünglichen Algorithmus erhalten.

6.6 Skalierung und mehrere Sequenzen

Für die in Abschnitt 6.2 eingeführten erweiterten Forward- und Backward-Variablen $\alpha_t(i)$ und $\beta'_t(i)$ gelten natürlich die gleichen numerischen Betrachtungen, die wir zu Beginn des Abschnitts 2.4 angestellt haben und die dazu führten, dass wir in der Praxis tatsächlich nur mit skalierten Variablen arbeiten. Dazu wurde der ursprüngliche Algorithmus nur leicht erweitert und es zeigte sich, dass mit den skalierten Größen sowohl die Likelihood $P(O|\lambda)$ bzw. $L(O|\lambda)$ als auch alle Reestimierungsparameter berechnet werden kann.

Es liegt auf der Hand, dass sich die gleiche Skalierungsmethode unverändert auf das erweiterte Modell übertragen lässt: Ersetzen wir in allen Gleichungen (2.26) bis (2.31) des Abschnitts 2.4 die Übergangswahrscheinlichkeit a_{ij} durch a_{ilj}, wobei wieder $l = cl(O_{(t)})$, und alle β durch β', erhalten wir skalierte Forward- und Backward-Variablen $\hat{\alpha}_t(i)$ und $\hat{\beta}'_t(i)$ für das erweiterte HMM; die entsprechenden Beweise und Gleichungen lassen sich eins zu eins übertragen. Genauso kann der für die Praxis modifizierte Viterbi-Algorithmus aus Abschnitt 2.4, bei dem statt der Likelihood deren Logarithmus maximiert wird, auf den im vorigen Abschnitt beschriebenen Viterbi-Algorithmus für das erweiterte HMM angepasst werden.

Schließlich bleibt noch die Erweiterung der Reestimierungsformeln auf das Training mit mehreren Sequenzen. In Abschnitt 2.5 hatten wir die Beweisidee zu den dort zuvor hergeleiteten intuitiven Formeln skizziert und gesehen, dass dazu in dem ursprünglichen Konvergenzbeweis nur die Q-Funktion anpasst werden musste. Nach den Ausführungen in Abschnitt 6.4 ist klar, dass diese Anpassung für das erweiterte Hidden-Markov-Modell analog durchgeführt werden kann. Wie auch beim klassischen HMM erhalten wir als Resultat die im Zähler und Nenner um die Summe über die K Sequenzen erweiterten Reestimierungsformeln mit den skalierten Forward- und Backward-Variablen $\hat{\alpha}_t(i)$ und $\hat{\beta}'_t(i)$.

Die kompletten skalierten Reestimierungsformeln für das hier eingeführte erweiterte HMM mit Ausgabe-Klassen finden sich im Anhang A. Die Gleichungen gelten für das Training mit mehreren Sequenzen und sind sowohl für Modelle mit diskreten Ausgaben als auch für Modelle mit stetigen Ausgaben aufgelistet.

Kapitel 7

Anwendungen: Sparzahlungen eines Bausparkollektivs

Die in Kapitel 2 beschriebenen Algorithmen für das klassische HMM wurden ebenso wie das in Kapitel 4 entwickelte Clusterverfahren und die Erweiterungen bezüglich der gestutzten Normalverteilung und den bedingten Übergängen aus den Kapiteln 5 und 6 in Form einer HMM-Programmbibliothek umgesetzt. Damit sind wir in der Lage, die in Abschnitt 3.3 vorgeschlagene Modellierung von Spargeld-Zeitreihen einer Bausparkasse zu realisieren.

In den folgenden Abschnitten stellen wir einige Auswertungen einer solchen HMM-Modellierung vor. Wir beschreiben zunächst den verwendeten Datensatz und die verschiedenen Modelltypen, die wir eingesetzt haben. In den ersten Auswertungen betrachten wir die Abhängigkeit der Zielfunktion des Clusterverfahrens von verschiedenen Initialisierungen und prüfen die Aussagekraft der Cluster-Indizes. Anschließend diskutieren wir die Einflüsse der Modelltypen und der Clusteranzahl auf die Verteilungen der von den trainierten Modellen generierten Sequenzen. Zuletzt gehen wir kurz auf die entstehende Clusterstruktur der Trainingsdaten ein und vergleichen die Laufzeiten des Maximum-Likelihood-Verfahrens bei Variation der HMM-Architektur. Wir beenden das Kapitel mit einer Diskussion der Ergebnisse.

7.1 Datensatz und relevante statistische Verteilungen

Wir setzen bei den folgenden Auswertungen zum Trainieren der Modelle stets einen festen Datensatz O10K von 10 000 Sequenzen ein, die aus den Originaldaten einer Bausparkasse gewonnen wurden. Dabei wurden Verträge eines Tarifs aus verschiedenen Abschlussjahrgängen berücksichtigt, die ihre Sparphase durch Erreichen der Zuteilung vollständig abgeschlossen haben. Jede Sequenz enthält die jährlichen Spargeldeingänge in Prozent der Bausparsumme (SPE) eines Vertrags und weist maximal 13 SPE-Einträge plus ein zusätzliches Endsymbol Θ auf (vgl. Abschnitt 3.3.2).

Die Abbildungen 7.1 bis 7.3 zeigen verschiedene statistische Verteilungen des Datensatzes, die für die Modellierung der Spar-Sequenzen bzw. für das Bausparkollektiv wichtig sind (vgl. dazu

die Abschnitte 3.1 und 3.3.2). Neben den Trainingsdaten O10K sind zum Vergleich die Verteilungen von 10 000 Testsequenzen T10K aufgetragen, die unter den gleichen Bedingungen wie die Trainingsdaten ausgewählt wurden und fast identische statistische Kenngrößen wie diese aufweisen. Für Mittelwert und Standardabweichung verwenden wir im Folgenden die Abkürzungen MW und STD.

Längenverteilung (Abbildung 7.1): Die Länge einer Sequenz entspricht dem relativen Zeitpunkt der Zuteilung in Vertragsjahren nach Abschluss. Da die Sparer mit zugeteiltem Vertrag die gesamte Bausparsumme einfordern können, stellt der Zuteilungszeitpunkt für die Bausparkasse eine wichtige Größe dar. Unter der Länge T einer Sequenz verstehen wir im Weiteren die Anzahl der SPE-Einträge *ohne* das Endsymbol. Damit gilt für beide Datensätze O10K und T10K: $T_{max} = 13$.

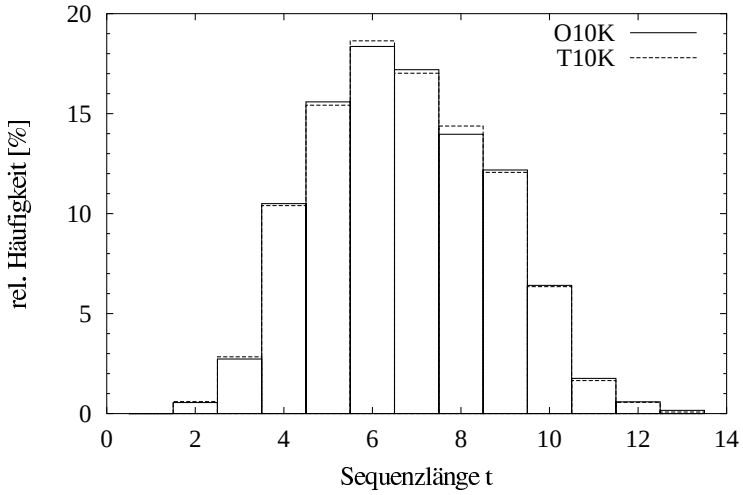


Abbildung 7.1: Längenverteilungen der Datensätze O10K (*MW:* 6.74, *STD:* 1.99) und T10K (*MW:* 6.72, *STD:* 1.97)

Spargeldeingänge pro Zeitraum (Abbildung 7.2): Die Sparleistungen eines Jahres fließen in die Zuteilungsmasse und beeinflussen damit die Liquidität der Bausparkasse. Als relevante Größen betrachten wir die Mittelwerte und die Standardabweichungen über die vorhandenen SPE-Einträge eines Zeitraums. In der Abbildung spiegelt sich das typische Verhalten der Sparer wider, von denen viele im ersten Vertragsjahr bereits die nötigen 40% SPE zum Erreichen des Mindestanspargrads einzahlen. Da andererseits auch viele Verträge zu Beginn noch gar nicht bespart werden, sind sowohl der mittlere SPE als auch die Standardabweichung in den ersten beiden Jahren maximal. Es fällt auf, dass die Werte der beiden Datensätze in den letzten Zeiträumen, in denen nur noch wenige Sequenzen vorhanden sind, stärker voneinander abweichen.

SPE-Summe einer Sequenz (Abbildung 7.3): Die Summe der Spargelder eines Vertrags bestimmt zusammen mit den jeweiligen Zinsen sein Guthaben bzw. den Anspargrad, also den Anteil des Guthabens an der Bausparsumme. Sofern nicht explizit etwas anderes gesagt wird, verstehen wir im Weiteren unter der SPE-Summe einer Sequenz die Summe ihrer SPE-Einträge. Zum Erstellen der SPE-Summen-Verteilung bilden wir die relativen

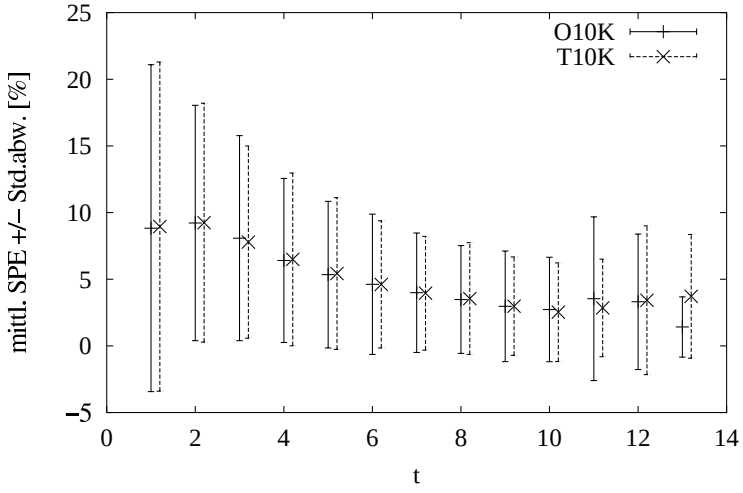


Abbildung 7.2: SPE der Datensätze O10K und T10K pro Zeitraum t ($MW \pm STD$)

Häufigkeiten der SPE-Summen in den 50 Intervallen $[0,2), [2,4), \ldots, [96,98), [98,\infty)$. In der Abbildung fällt auf, dass keine Sequenz eine kleinere SPE-Summe als 34% aufweist und fast ein Viertel aller Sequenzen in der Klasse landen, die einer SPE-Summe von 38% – 40% entspricht. Der Grund liegt in dem tariflichen Mindestanspargrad von 40%, in den neben den Sparzahlungen noch die Zinsen eingehen und der damit eine wichtige deterministische Nebenbedingung für die Sequenzen darstellt (vgl. Abschnitt 3.3.2).

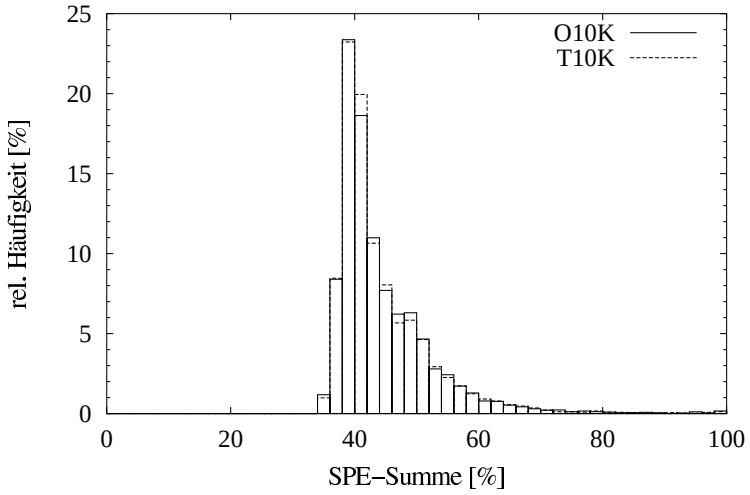


Abbildung 7.3: SPE-Summen der Datensätze O10K (*MW:* 44.45%, *STD:* 7.89%) und T10K (*MW:* 44.36%, *STD:* 7.71%); Intervallbreite 2%

Wie bereits in Abschnitt 4.3.4 angesprochen sollen diese Verteilungen zusammen mit den Maßzahlen Mittelwert und Standardabweichung der Längen- und SPE-Summen-Verteilung als Referenzen dienen, wenn wir mit trainierten Hidden-Markov-Modellen Sequenzen erzeugen. Es sei noch darauf hingewiesen, dass weniger als 10% aller jährlich abgeschlossenen Verträge des

Tarifs, aus dem die Daten stammen, länger als 13 Jahre bis zur Zuteilung benötigen, d. h. die Sequenzen der Datensätze O10K und T10K decken im Wesentlichen das typische Sparverhalten der Verträge ihres Tarifs ab, die bis zur Zuteilung gelangen.

7.2 Architektur der verwendeten Modelle

Wir stellen die verschiedenen Komponenten Modelltopologie, Verteilungsfunktionen und Modelltyp der HMM-Architektur vor, die für die Auswertungen dieses Kapitels eingesetzt und gegebenenfalls kombiniert wurden. Die Begriffe Topologie und Struktur werden wieder synonym verwendet (vgl. Abschnitt 3.3.3).

Modelltopologien

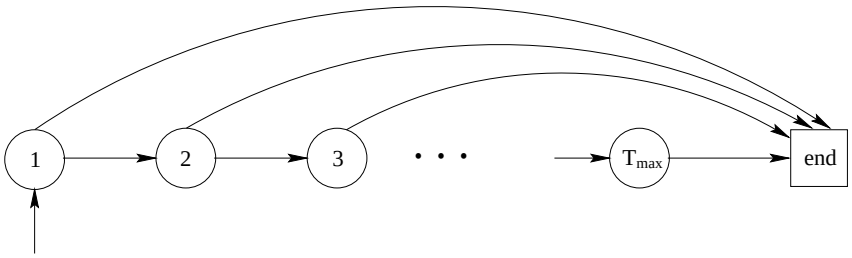
Zur Clusterung und zum Training der Sequenzen ziehen wir die Modelltopologien bzw. -strukturen heran, die in Abbildung 7.4 zu sehen sind. Wir unterscheiden dabei den Endzustand, der nur das Endsymbol ausgibt und bei dem die Markov-Kette definitiv endet, von den anderen Zuständen, die wir auch als SPE-Zustände bezeichnen. Die Modellstrukturen wurden zum Teil bereits in Abschnitt 3.3.3 vorgestellt und besitzen im Einzelnen folgende Eigenschaften:

SLRMIN – minimales striktes Links-Rechts-Modell:
Ein solches Modell mit $T_{max} = 13$ SPE-Zuständen stellt das kleinste Links-Rechts-Modell dar, das alle Sequenzen des Datensatzes O10K abbilden kann. Jeder Zustand entspricht genau einem Zeitraum *t*, da alle Sequenzen im ersten Zustand beginnen und nur Übergänge zum jeweiligen direkten Nachfolger oder zum Endzustand vorhanden sind.

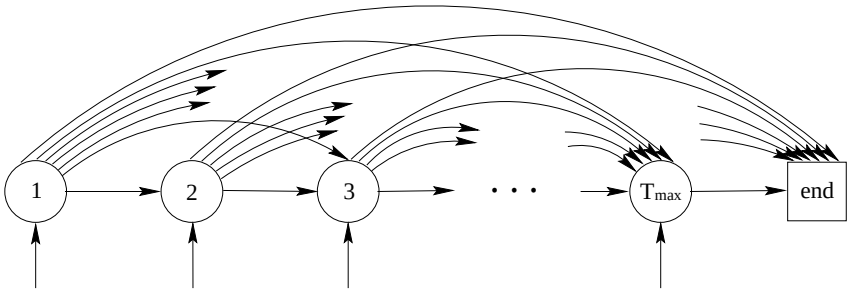
SLRMAX – maximales striktes Links-Rechts-Modell:
Bei fester Anzahl von Zuständen sind bei dieser Topologie die Übergänge zu allen Nachfolgezuständen vorhanden, deshalb nennen wir es maximal. Wie bei der Modelltopologie SLRMIN können hier nur Sequenzen der Länge *S* modelliert werden, wenn die Topologie *S* SPE-Zustände hat. Allerdings entfällt hier die Äquivalenz von einem Zustand und einem festen Zeitraum *t*, da die Sequenzen in allen Zuständen „starten“ und beliebig viele Zustände überspringen können.

LRMIN – „minimales“ Links-Rechts-Modell:
Diese Topologie entspricht der um Selbstübergänge und Startwahrscheinlichkeiten für jeden Zustand erweiterten Struktur SLRMIN. Die Motivation eines solchen Modellgraphen liegt in der Tatsache, dass es in einem Bausparkollektiv viele Regelsparer gibt, die jedes Jahr ähnlich hohe Beträge einzahlen. Aufgrund der Selbstübergänge in dem Graphen können mit einer solchen Topologie beliebig lange Sequenzen modelliert werden.

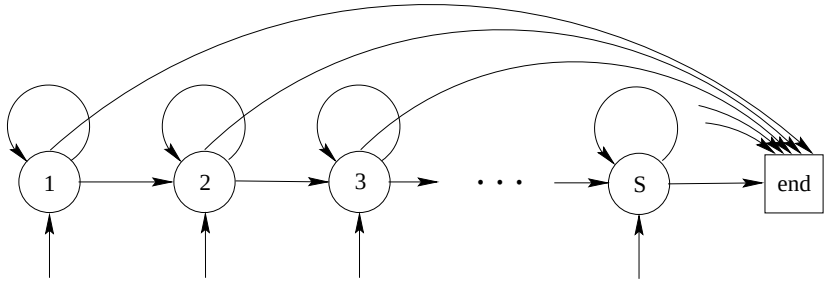
VG – vollständiger Modellgraph:
Als letzte Variante betrachten wir schließlich eine Toplogie mit einem vollständigen Graphen, bei dem es von jedem SPE-Zustand eine Verbindung zu allen anderen Zuständen gibt. Mit dieser Modellstruktur können ebenfalls beliebig lange Sequenzen abgebildet werden.



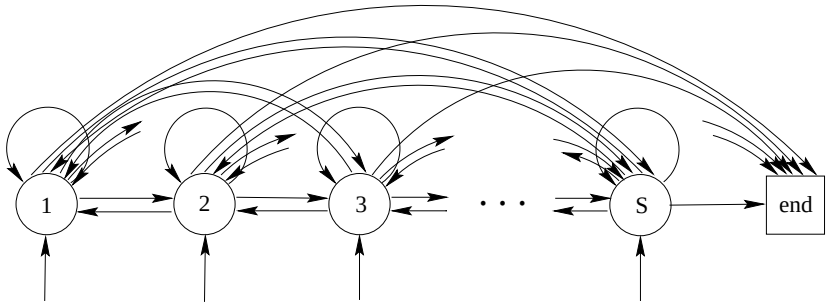
SLRMIN(T_{max})



SLRMAX(T_{max})



LRMIN(S)



VG(S)

Abbildung 7.4: Modelltypen für die Spargeld-Sequenzen; die Werte in den Klammern geben die Anzahl der jeweiligen SPE-Zustände an.

Allen vier Modelltopologien ist gemein, dass von jedem Zustand ein Übergang in den Endzustand besteht. Damit wird die Abbildung jeder Sequenz gewährleistet, die kürzer als T_{max} ist.

In den folgenden Abschnitten werden wir uns auf die Modellstrukturen **SLRMIN(13)**, **SLRMAX(13)**, **LRMIN(6)** und **VG(6)** konzentrieren. Die 13 SPE-Zustände bei den strikten Links-Rechts-Modellen ergeben sich aus den oben genannten Gründen, während wir die Anzahl der SPE-Zustände bei den beiden Topologien mit Kreisen bewusst kleiner wählen, um die Anzahl der freien Parameter zu reduzieren. In Tabelle 7.1 wird die unterschiedliche Größe der so gewählten Topologien deutlich.

Topologie	SLRMIN(S)	SLRMAX(S)	LRMIN(S)	VG(S)
Übergänge	$2S-1$	$\frac{1}{2}S(S+1)$	$3S-1$	$S(S+1)$
Startwahrsch.	1	S	S	S
	$S=13$	$S=13$	$S=6$	$S=6$
Übergänge	25	91	17	42
Startwahrsch.	1	13	6	6

Tabelle 7.1: Anzahl der Kanten bzw. der positiven Einträge von A und π bei den verschiedenen Modellgraphen

Verteilungsfunktionen der Ausgaben

Als Dichtefunktion auf den Ausgaben setzen wir im Wesentlichen die stetige Dichte der linksseitig bei $\varepsilon = -0.1$ gestutzten Normalverteilung ein, für die wir in Kapitel 5 die Reestimierungsformeln des HMM-Trainings angepasst haben (vgl. Abschnitt 7.5.1).

Wir arbeiten bei allen Auswertungen jeweils mit nur einer Mischkomponente, so dass die Ausgabeverteilung eines Zustands nur ein einziges Maximum aufweist. Dies hat den Hintergrund, dass wir eine Überlagerung von mehreren Häufigkeitspunkten im Spargeldeingang bereits über die Clusterung abbilden können. Zudem erhöht eine größere Anzahl von Mischkomponenten die Komplexität der Modelle; wir bevorzugen jedoch in unseren Anwendungen, bei denen durch die trainierten Modelle typisches und ähnliches Sparerverhalten klassifiziert werden soll, eher einfache Modelle bei gegebenenfalls höherer Clusteranzahl.

Auf die Verwendung von diskreten Ausgabeverteilungen verzichten wir aus den in Abschnitt 3.3.3 genannten Gründen.

HMM-Typen

In den folgenden Anwendungen wird unter anderem das erweiterte HMM mit bedingten Übergangswahrscheinlichkeiten eingesetzt und mit dem klassischen HMM verglichen. Wir unterscheiden deshalb die beiden HMM-Typen **KL-HMM** (klassisches HMM) und **AK6-HMM** (erweitertes HMM mit 6 Ausgabe-Klassen). Die Ausgabe-Klassen der erweiterten Modelle entsprechen dabei einer Einteilung der Ausgabesummen einer (Teil-)Sequenz in folgende sechs Intervalle:

$$[0,12),[12,24),[24,36),[36,48),[48,60),[60,\infty)\,.$$

Bei Einsatz eines solchen erweiterten HMM werden aus jeder Kante in den vorgestellten Modelltopologien sechs bedingte Übergangskanten mit individuellen Wahrscheinlichkeiten, die nur mit den Daten trainiert werden, deren bisherige Ausgabesumme beim Verlassen des Zustands in dem jeweiligen Intervall liegt (vgl. Abschnitte 6.1 und 6.3).

Da die Spargeldeingänge der Sequenzen in Prozent der Bausparsumme vorliegen, ist das letzte Intervall bei den Trainingsdaten dünn belegt, und keine Sequenz weist eine höhere SPE-Summe als 100% auf (vgl. Abbildung 7.3).

7.3 Initialisierungen für das Clusterverfahren

Wir vergleichen zunächst die in Abschnitt 4.2.3 beschriebenen Möglichkeiten zur Initialisierung des Maximum-Likelihood-Verfahrens, das wir in Kapitel 4 als HMM-basiertes Clusterverfahren vorgestellt hatten. Wir beschränken uns dabei auf die folgenden drei Varianten:

1. **Individuelle Startmodelle:** Jedes HMM wird individuell unter Einhaltung der stochstischen Nebenbedingungen mit Zufallsparametern belegt, wobei die Parameter der Dichtefunktionen innerhalb vorgegebener Schranken liegen müssen, die wir unter Berücksichtigung der Daten wählen ($\mu \in [0, 50]$, $\sigma \in [7, 55]$). Das Clusterverfahren startet mit der Zuordnung der Sequenzen zu den jeweils besten Modellen.
2. **Zufallspartition:** Die Start- und Übergangswahrscheinlichkeiten aller Modelle werden gleichförmig initialisiert, wobei die Parameter der Dichtefunktion in den Bereichen $\mu \in [10, 30]$ und $\sigma \in [45, 55]$ liegen, so dass jedes Modell grundsätzlich jede der Trainingssequenzen darstellen kann. Das Clusterverfahren startet mit einer zufälligen Partitionierung der Sequenzen, d. h. jedes HMM wird mit einem der Zufallscluster trainiert.
3. ***K*-means-Partition:** Die Modelle werden wie bei der Zufalls-Partition initialisiert, hier jedoch mit den nach dem *K*-means-Verfahren vorsortierten Daten trainiert (vgl. Abschnitt 3.2.3).

Abbildung 7.5 vergleicht für den Modelltyp SLRMIN(13) die aus den Clusterungen resultierenden Likelihoodwerte (siehe Gleichung (4.2)) bei verschiedenen Initialisierungen und variierender Clusteranzahl *K*. Für jedes *K* sind dabei der Mittelwert (MW) und das Maximum (Max) aus je fünf Zufallsbelegungen aufgetragen.

Es zeigt sich, dass eine individuelle Zufallsbelegung der Startmodelle in allen Fällen schlechter abschneidet als die anderen Initialisierungsformen. Der Grund liegt vermutlich darin, dass die Startmodelle für vorsortierte Daten allgemeiner sind und sich dadurch im ersten Trainingsschritt bereits gut den Daten anpassen können, während zufällige Startmodelle bereits spezielle Bereiche des Datenraums abdecken und dadurch weniger gut auf alle Sequenzen trainiert werden können.

Für das strikte Links-Rechts-Modell ergeben sich bei einer *K*-means-Gruppierung der Sequenzen etwas bessere Werte als bei einer zufälligen Partitionierung. Dies lässt sich dadurch erklären, dass bei einer SLRMIN-Topologie jeder Zustand genau einem Zeitraum in den Sequenzen entspricht und deshalb das HMM umso besser trainiert werden kann, je näher die SPE-Einträge

jedes Jahres beieinander liegen. Das *K*-means-Verfahren minimiert aber gerade die Abstände dieser jährlichen Einträge.

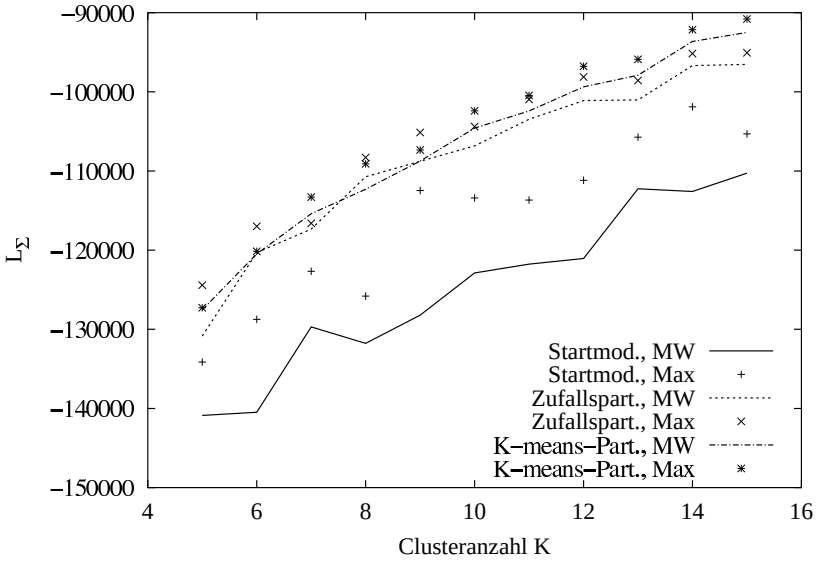


Abbildung 7.5: SLRMIN(13), Vergleich der Likelihoodwerte bei verschiedenen Initialisierungen (*Daten O10K; Mittelwert und Maximum aus je 5 Clusterungen pro K*)

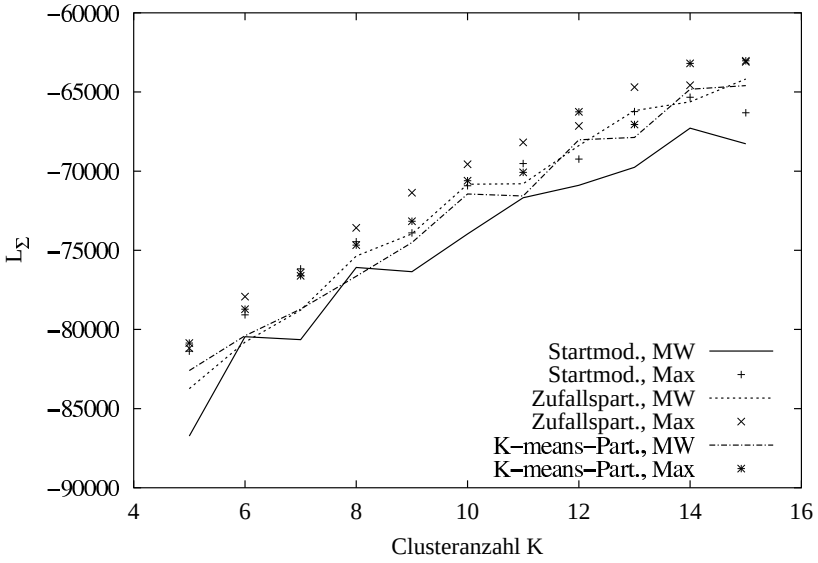


Abbildung 7.6: VG(7), Vergleich der Likelihoodwerte bei verschiedenen Initialisierungen (*Daten O10K; Mittelwert und Maximum aus je 5 Clusterungen pro K*)

Bei Verwendung der Topologie VG(6) sind die Unterschiede der drei Initialisierungsmethoden nicht so ausgeprägt (siehe Abbildung 7.6), wobei jedoch auch hier die Zufallsbelegung der Startmodelle i. d. R. etwas schlechter abschneidet. Erwartungsgemäß liefert hier die *K*-means-Vorpartitionierung keine besseren Werte als eine Zufallspartition, denn der oben genannte struktu-

relle Vorteil entfällt bei einem VG-Modell.

Alle weiteren Clusterungen dieses Kapitels wurden mit zufälligen Sequenzpartitionen durchgeführt, da das *K*-means-Verfahren die Gesamtlaufzeit zusätzlich erhöht und nur bei speziellen Modellen etwas bessere Zielfunktionswerte liefert.

7.4 Cluster-Indizes

In Abschnitt 4.3 wurden verschiedene Aspekte zur Bewertung von Clusterergebnissen des Maximum-Likelihood-Verfahrens angesprochen und verschiedene Gütemaße vorgestellt, die anhand eines stark strukturierten Testdatensatzes von 500 Sequenzen mit jeweils zwei Einträgen und unter Einsatz zweier kleiner Hidden-Markov-Modelle erläutert wurden. Während die Likelihoodsumme $L_\sum$ keine Rückschlüsse auf eine geeignete oder den Daten angepasste Clusteranzahl zuließ, war dies bei den Testdaten über die von uns entwickelten Indizes durchaus möglich (vgl. Abschnitte 4.3.1 bis 4.3.3).

Bei Verwendung der weniger strukturierten Daten O10K und der Modelltopologien von Seite 95, mit denen wesentlich flexiblere Modelle trainiert werden, zeigen sich jedoch schnell die Grenzen dieser Indizes. In allen durchgeführten Clusterungen lieferte das Abstandsmaß $D_s(\lambda_1, \lambda_2)$ zwischen zwei Modellen λ_1 und λ_2 *immer* den Wert ∞ zurück, da von einem der beiden Modelle regelmäßig eine Sequenz erzeugt wurde, die vom anderen Modell nicht abgebildet werden konnte bzw. deren Likelihood $L(O|\lambda) = 0$ ergab. Damit wird aber der DB-Index aus Gleichung (4.5), der auf diesem Maß beruht, bei unseren Anwendungen grundsätzlich auf null gesetzt.

Die Indizes *KS*, *CS* und *ID* aus den Gleichungen (4.7), (4.10) und (4.11), die die durchschnittliche Anpassung der trainierten Ausgabe-Verteilungen der Zustände bewerten, lassen sich dagegen bestimmen. In den nächsten beiden Abschnitten ziehen wir zur Bewertung von Variationen der Anzahl von Clustern und Zuständen jedoch nur den ID-Index heran, da sowohl der KS-Index als auch der CS-Index bei unseren Auswertungen keine brauchbaren Ergebnisse lieferten: Beide Größen hängen zu stark von den Initialparametern ab, als dass wir konkrete Schlüsse aus dem Verlauf der Kurven ziehen könnten.

7.4.1 Variation der Clusteranzahl

Der ID-Index entspricht der über alle Zustände und alle Modelle gemittelten Differenz der Datenverteilung und der analytischen Verteilung eines Zustands (vgl. Abschnitt 4.3.3). Er beinhaltet zwei gegenläufige Effekte: Einerseits können sich die Hidden-Markov-Modelle bei Erhöhung der Clusteranzahl besser auf die Daten spezialisieren und damit die mittlere Integraldifferenz *ID* verringern; andererseits werden dann den Modellen im Schnitt immer weniger Daten zugeordnet, d. h. die mittlere Anzahl der an dem Training der Ausgabe-Verteilungen beteiligten Daten sinkt. Wie wir in Abschnitt 4.3.3 festgestellt hatten, ist aber unter der Annahme, dass die gleiche Übereinstimmung der Daten mit der analytischen Verteilung vorliegt, die Integraldifferenz der

Verteilungen umso größer, je weniger Daten an der empirischen Verteilung beteiligt sind. Damit könnte der ID-Index bei Erhöhung der Clusteranzahl durchaus in einem globalen Minimum landen, in dem sich die beiden gegenläufigen Effekte optimal ausgleichen.

Für die Topologien SLRMIN(13) und LRMIN(6) bei Verwendung der Modelltypen mit Ausgabe-Klassen wurde deshalb sukzessive die Clusteranzahl K erhöht, und für jedes K wurden 10 Clusterungen des Datensatzes O10K mit zufälligen Startpartitionen durchgeführt. In Abbildung

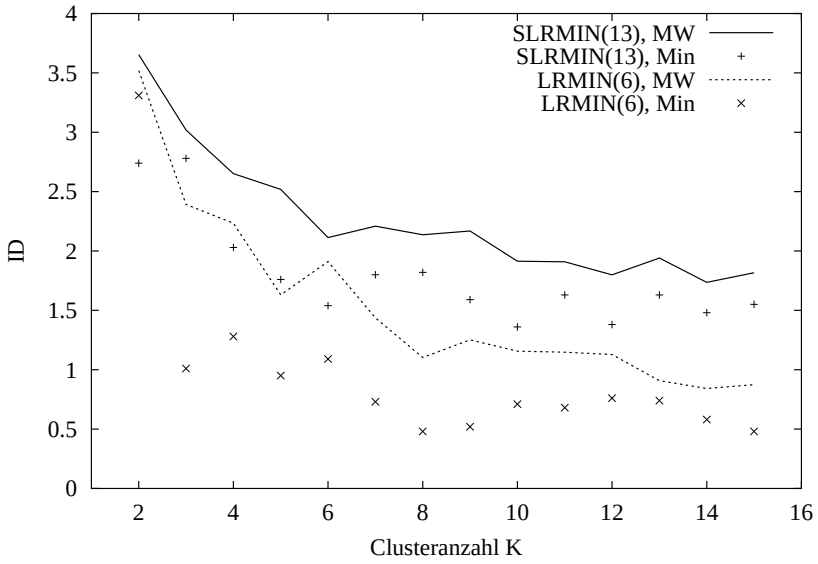


Abbildung 7.7: ID-Indizes für die AK6-Modelle SLRMIN(13) und LRMIN(6) (*Daten O10K; Mittelwert und Minimum bei jeweils 10 Clusterungen pro K*)

7.7 sind die resultierenden Mittelwerte und Minima des ID-Indexes aufgetragen. Am großen Abstand zwischen den Mittelwerten und den Minimalwerten ist zu erkennen, wie stark der ID-Index von der Initialisierung der Clusterung abhängt. Der Verlauf der Mittelwerte zeigt, dass bei beiden Modelltypen die Intergraldifferenzen zwischen den empirischen und den trainierten analytischen Ausgabe-Verteilungen mit steigender Clusteranzahl geringer werden und dass die Werte der Modellstruktur LRMIN(6) mit der größeren Anzahl von freien Parametern grundsätzlich besser sind als bei der Struktur SLRMIN(13). Obgleich die lokalen Minima bei 6 und 8 Clustern bei eventuell gewünschter Einschränkung der Modellanzahl als Hinweis auf eine geeignete Clusteranzahl interpretiert werden können, erscheint uns die Aussagekraft des ID-Indexes bezüglich einer optimalen Clusteranzahl angesichts dieser Kurven jedoch relativ schwach.

An dieser Stelle wird ein Problem deutlich, das auch schon bei den früheren Simulationsmodellen für Bausparkollektive aufgetreten ist: Die Daten der Sparer haben zwar bestimmte ausgeprägte Häufungspunkte, sind aber im Allgemeinen nicht stark strukturiert und in klar zu bestimmenden Gruppen voneinander abgrenzbar (vgl. [34, 44]).

7.4.2 Variation der Anzahl der Zustände

Die zu Beginn des vorigen Abschnittes erwähnten gegenläufigen Effekte des ID-Indexes gelten auch bei Erhöhung der Anzahl der Zustände, wenn gleichzeitig die Anzahl der Modelle bzw. Cluster festgehalten wird. Es sollte deshalb getestet werden, ob anhand des ID-Indexes eine Aussage über die benötigte Anzahl von Zuständen in einem Modell getroffen werden kann.

Wir führten unter Verwendung von Modellen mit Ausgabe-Klassen und der Topologie LRMIN eine Reihe von Clusterungen der Trainingsdaten O10K durch, wobei die Anzahl der SPE-Zustände in den Modellen nach und nach erhöht, die Clusteranzahl $K = 15$ aber festgehalten wurde. Nach jeder Clusterung wurde für die trainierten Modelle sowohl mit den Daten O10K als auch mit den Testdaten T10K der ID-Index bestimmt.

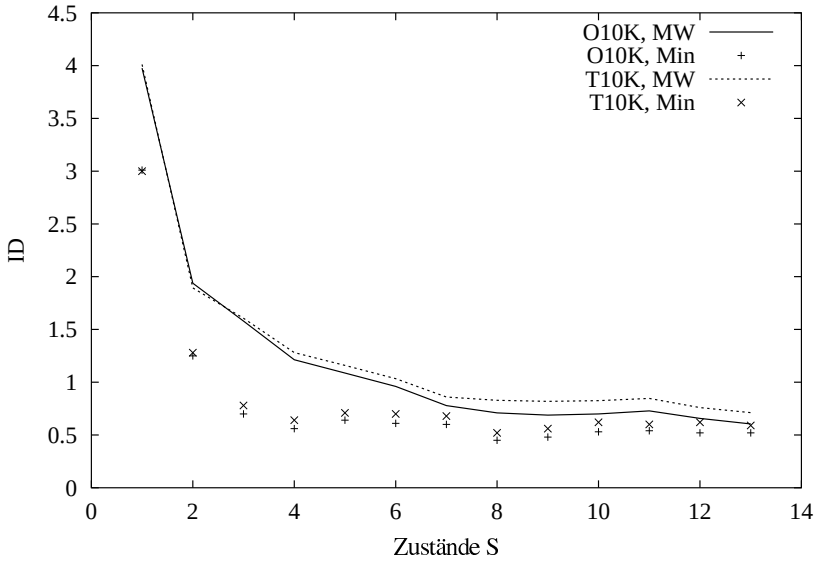


Abbildung 7.8: ID-Indizes des AK6-Modells LRMIN(S) bei variierender Anzahl S von Zuständen für die Daten O10K und T10K (*Clusteranzahl* $K = 15$ *fest; Modelltraining mit den Daten O10K; Mittelwert und Minimum bei jeweils 10 Clusterungen pro* S)

In Abbildung 7.8 ist zu sehen, dass der über jeweils 10 Läufe gemittelte Index bei 9 SPE-Zuständen ein lokales Minimum annimmt. Daneben ist bei wachsender Anzahl von Modellzuständen eine zunehmende Spezialisierung der Modelle auf den Datensatz O10K zu erkennen, denn die die beiden Kurven von Trainings- und Testdaten laufen für größere S auseinander. Allerdings schwanken auch hier die Werte wieder stark bei unterschiedlicher Initialisierung.

Die ID-Kurven sprechen dafür, dass bezüglich der Anpassung der Verteilungen an die Daten der Zustände LRMIN-Strukturen mit 8 oder 9 Zuständen ausreichen. Allerdings ist auch hier die Aussage recht schwach.

7.5 Generierung von Sequenzen

Das Hauptziel der Clusterung von Spargeld-Sequenzen besteht darin, geeignete Modelle zu finden, die das statistische Verhalten der Daten widerspiegeln und die für das Generieren von Sequenzen zur Simulation einer zukünftigen Entwicklung eingesetzt werden können. Wir testen deshalb in den folgenden Abschnitten die trainierten Modelle, die das Ergebnis einer Clusterung mit dem Datensatzes O10K darstellen, indem wir mit ihnen Sequenzen erzeugen und ihre Verteilungen mit denen der Trainingsdaten vergleichen. Dabei betrachten wir neben der Variation der Clusteranzahl und der Modelltopologie vor allem die Auswirkungen des verwendeten Modelltyps auf die Zusammensetzung der generierten Daten.

Für die Vergleiche erzeugen wir grundsätzlich mit jedem HMM genauso viele Sequenzen, wie während des Clusterverfahrens dem Modell zugeordnet wurden und mit denen das HMM zuletzt trainiert wurde, so dass wir immer auf einen Datensatz von ebenfalls 10 000 Sequenzen kommen.

7.5.1 Klassisches HMM (KL-HMM)

Nachdem wir uns in Kapitel 5 ausführlich mit der gestutzen Normalverteilung beschäftigt haben, betrachten wir zunächst, wie sich diese im Vergleich zu der auf der x-Achse unbeschränkten Normalverteilung auf die Simulation von Spargeld-Sequenzen auswirkt.

Vergleich der Normalverteilung mit der gestutzten Normalverteilung

Wir vergleichen zwei Clusterungen mit jeweils 25 Clustern, wobei wir einmal Modelle mit gestutzten Dichten und einmal solche mit in x-Richtung unbeschränkte Dichten der Normalverteilung vorgeben und entsprechend trainieren. Die hohe Anzahl von 25 Clustern wählen wir deshalb, um eine schlechte Anpassung aufgrund von zu wenigen Modellen weitgehend auszuschließen.

In Abildung 7.9 sind die relativen Häufigkeiten der Spargeldeingänge der Originalsequenzen im ersten Zeitraum im Vergleich mit denen der generierten Sequenzen aus den trainierten Modellen der beiden Clusterungen zu sehen. Die SPE-Einträge wurden dazu jeweils in äquidistante Klassen der Breite 2% einsortiert. Es zeigt sich, dass die Sequenzen der Modelle mit gestutzten Normalverteilungen die Verteilung der Originaldaten im vorderen Bereich, der am stärksten belegt ist, sehr gut treffen, während bei den Modellen mit der gewöhnlichen Normalverteilung sehr viele Ausgaben im negativen Bereich entstehen (Anteil ca. 10%), so dass die Gesamtverteilung der Trainingsdaten wesentlich schlechter approximiert wird. Die Spitze im Bereich von 40% bei den Daten O10K wird dagegen von beiden generierten Spargeld-Verteilungen nur leicht und mit breiterer Streuung wiedergegeben.

Im Vergleich der beiden Verteilungen schneidet die gestutzte Normalverteilung auch bei Einsatz von anderen Modellstrukturen und Clustervorgaben besser ab, und folglich verwenden wir zur Simulation von positiven Spargeld-Sequenzen weiterhin für beide Modelltypen KL-HMM und AK-HMM ausschließlich diese Verteilung.

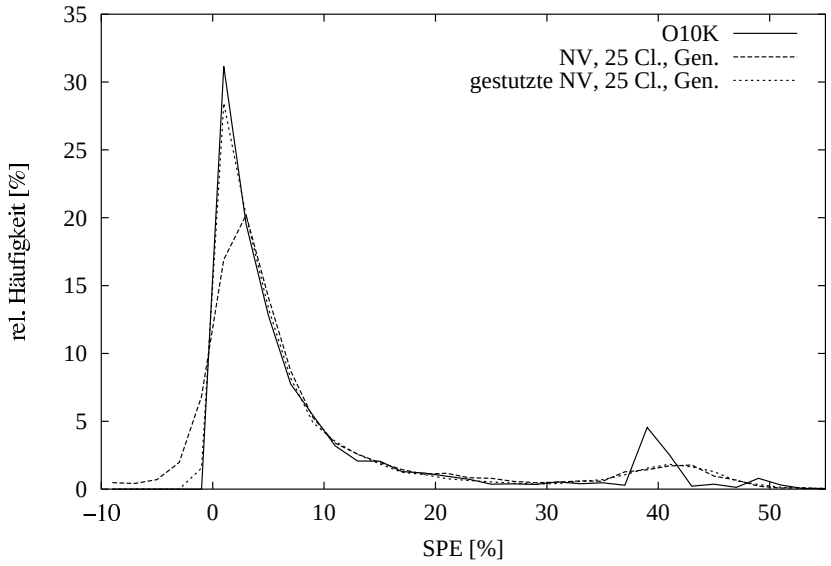


Abbildung 7.9: SPE-Verteilung im 1. Jahr bei Trainingsdaten und generierten Daten von Modellen mit Normalverteilungen (NV) und gestutzten NV (*SLRMIN, KL-HMM, jeweils 25 Cluster*)

Modelltopologie und Clusteranzahl

Wir vergleichen nun die generierten Sequenzen von trainierten Modellen aus jeweils zwei verschiedenen Clusterungen ($K = 3$ und $K = 9$) mit den Trainingsdaten O10K in den Abbildungen 7.10 bis 7.13.

Jede der vier Abbildungen entspricht einer der vier Modellstrukturen aus Abbildung 7.4 bei Verwendung des Modelltyps des klassischen HMM. Den beiden dargestellten Verteilungen liegen jeweils diskrete Werte zugrunde, wobei die SPE-Summen wieder in äquidistante Intervalle der Breite 2% aufgeteilt wurden (vgl. Bemerkung S. 92 sowie Abbildungen 7.1 und 7.3); die Liniendarstellung wurde aus Gründen der Unterscheidbarkeit der verschiedenen zusamengehörigen Werte gewählt. Zu jeder Verteilung wurden der Mittelwert und die Standardabweichung berechnet und als Fehlerbalken (MW $\pm$ STD) in die entsprechende Graphik integriert.

Wenn wir die Längenverteilungen betrachten, fällt auf, dass die Sequenzen der beiden strikten Links-Rechts-Modelle SLRMIN(13) und SLRMAX(13) sehr gut die Längen der Trainingsdaten widerspiegeln (Abbildungen 7.10 und 7.11), während bei den anderen Modellen durch die Kreise in den versteckten Markov-Ketten sowohl wesentlich längere als auch kürzere Sequenzen entstehen, was sich auch in den größeren Standardabweichungen niederschlägt (Abbildungen 7.12 und 7.13). Durch Erhöhung der Clusteranzahl lassen sich diese Werte zwar verbessern, aber die rechten Enden der Verteilungen bleiben stark ausgeprägt. Die Mittelwerte der generierten Sequenzlängen stimmen dagegen erfreulicherweise bei allen Modelltypen fast exakt mit der mittleren Sequenzlänge der Originaldaten überein.

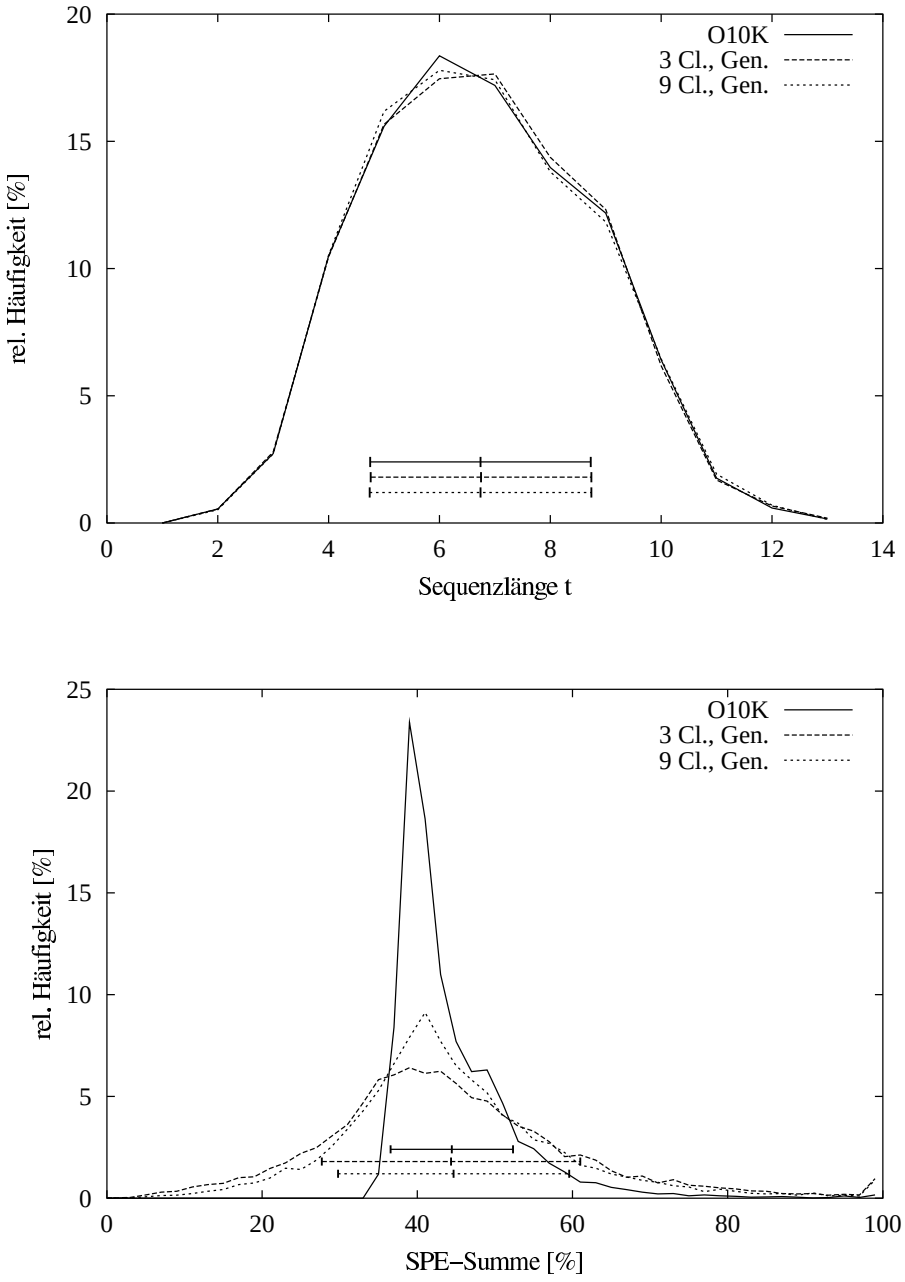


Abbildung 7.10: SLRMIN(13), KL-HMM; Verteilungen der Trainingsdaten O10K und 10 000 generierter Daten bei verschiedenen Clusterungen (*Fehlerbalken: MW* $\pm$ *STD*)

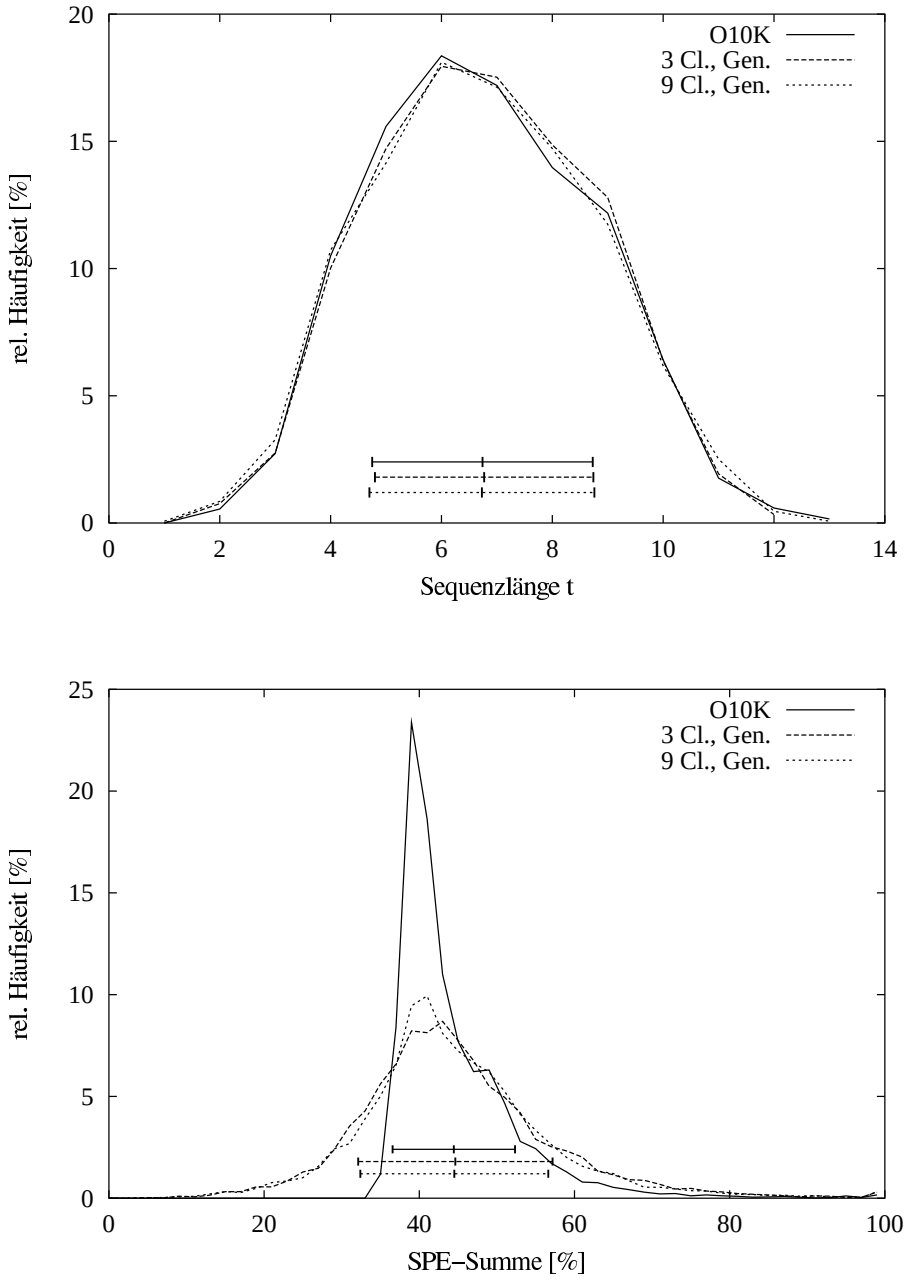


Abbildung 7.11: SLRMAX(13), KL-HMM; Verteilungen der Trainingsdaten O10K und 10 000 generierter Daten bei verschiedenen Clusterungen (*Fehlerbalken: MW* $\pm$ *STD*)

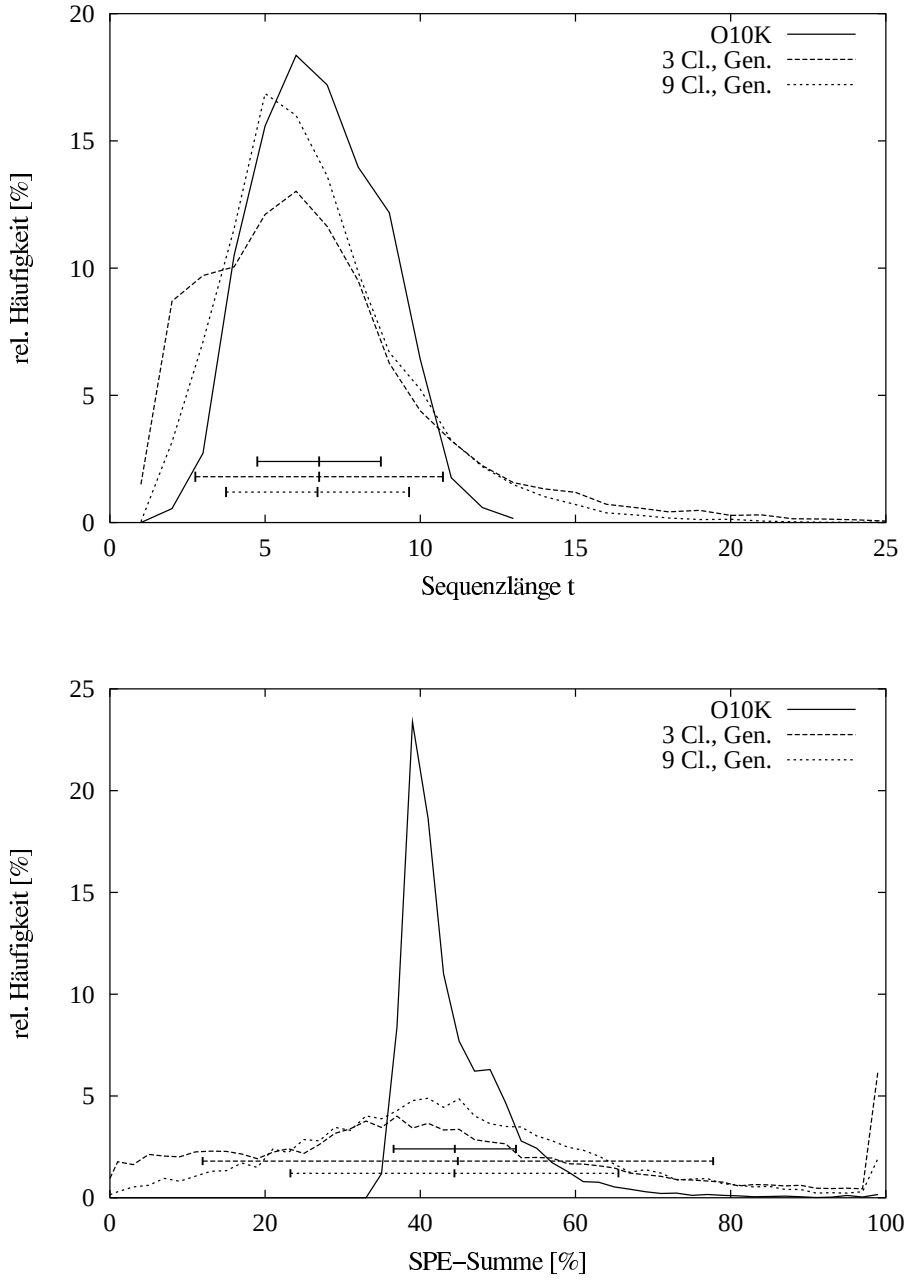


Abbildung 7.12: LRMIN(6), KL-HMM; Verteilungen der Trainingsdaten O10K und 10 000 generierter Daten bei verschiedenen Clusterungen (*Fehlerbalken: MW* $\pm$ *STD*)

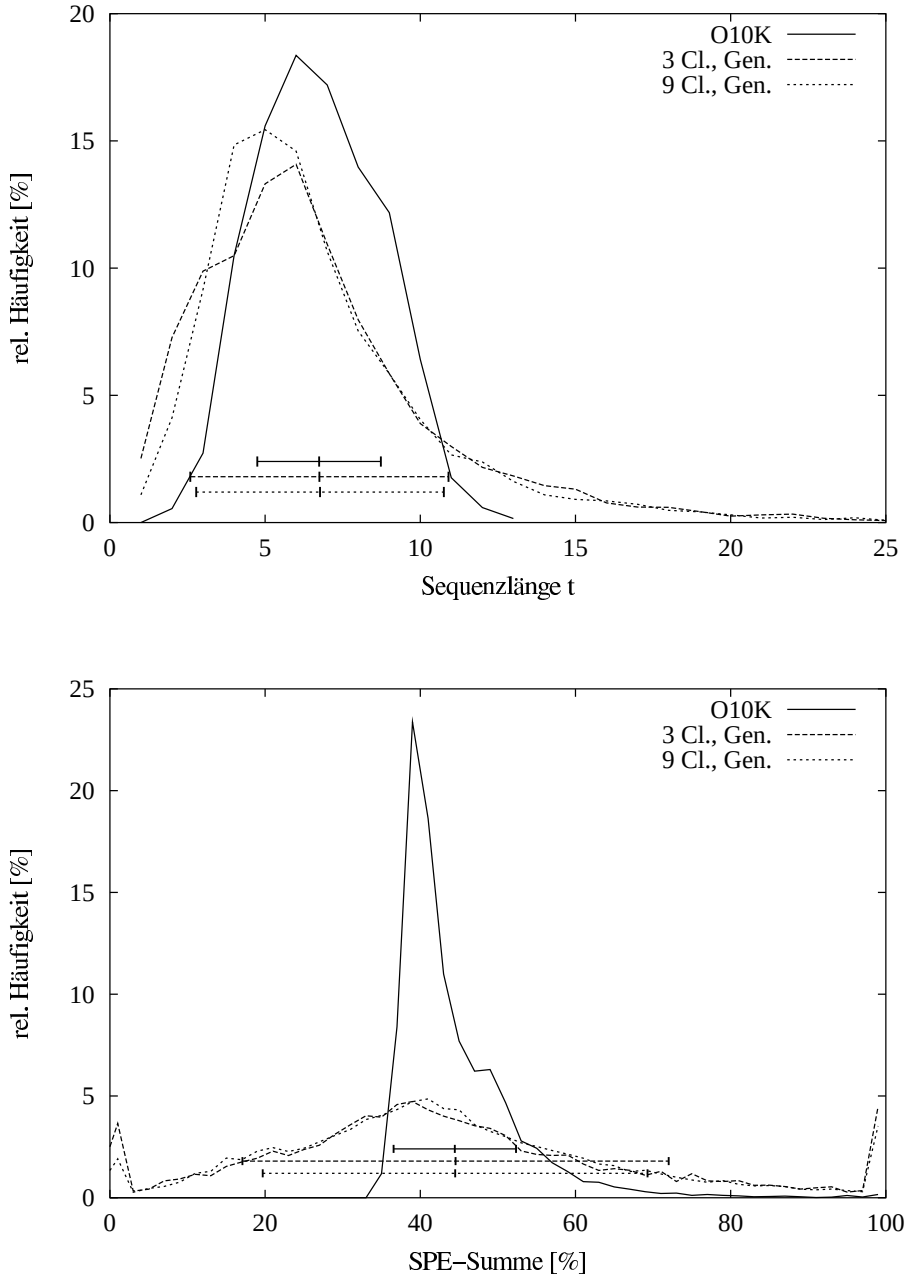


Abbildung 7.13: VG(6), KL-HMM; Verteilungen der Trainingsdaten O10K und 10 000 generierter Daten bei verschiedenen Clusterungen (*Fehlerbalken: MW* $\pm$ *STD*)

Die in den jeweils unteren Bildern aufgetragenen relativen Häufigkeiten der SPE-Summen zeigen bei den generierten Sequenzen grundsätzlich einen anderen Verlauf als bei den Originaldaten. Während sich dort die meisten SPE-Summen im Bereich um den Mindestanspargrad von 40% konzentrieren, produzieren die trainierten Modelle immer auch sehr viele Sequenzen, deren SPE-Summe sehr groß oder sehr klein ist, so dass die Enden der Verteilungen wesentlich ausgeprägter sind als die Enden bei der Verteilung der Trainingsdaten. Die Folge davon sind die sehr großen Standardabweichungen der SPE-Summen. Die Modelle der Topologien LRMIN(6) und VG(6) schneiden hier noch schlechter ab als die strikten Links-Rechts-Modelle. Auffällig sind hier vor allem die hohen Anteile der SPE-Summen in den äußeren Klassen $[0, 2)$ und $[98, \infty)$ (Abbildungen 7.12 und 7.13). Die Mittelwerte sind jedoch für alle Modellstrukturen auch bei den SPE-Summen fast gleich dem Mittelwert der SPE-Summe bei den Daten O10K.

Eine Erhöhung der Clusteranzahl bzw. der Anzahl der zum Training der Daten verwendeten Modelle von 3 auf 9 führt besonders bei den Modelltopologien LRMIN(6) und VG(6) zu besseren Ergebnissen. Das generelle Problem der zu großen Streuungen bei diesen Strukturen bleibt jedoch bestehen. Weitere Untersuchungen mit größerer Clusteranzahl K bei allen Topologien brachten ebenfalls keine signifikanten Verbesserungen der Längen- und der SPE-Summen-Verteilungen.

Schließlich betrachten wir noch für die beiden Modellstrukturen SLRMIN(13) und LRMIN(6) die mittleren Einträge und die Standardabweichungen des Spargeldeingangs in jedem Zeitraum t bei einer Clusterung mit jeweils 9 Modellen und vergleichen diese mit denen der Originaldaten (Abbildung 7.14). Bis auf die letzten Jahre, in denen aber die Datenbasis sehr dünn ist und auch nur wenige Sequenzen von den trainierten Modellen erzeugt werden, passen die Werte

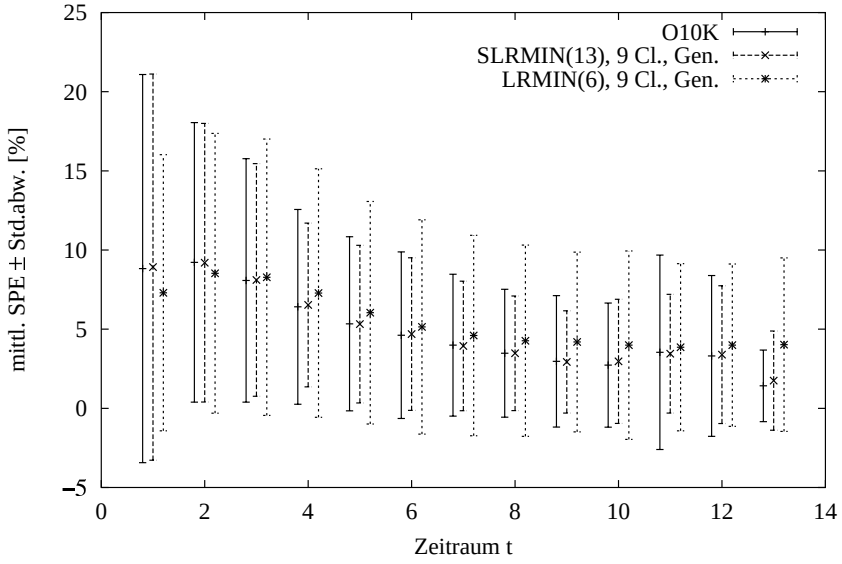


Abbildung 7.14: SLRMIN(13) und LRMIN(6), KL-HMM; SPE pro Zeitraum t der Trainingsdaten O10K und 10 000 generierter Sequenzen (*MW* $\pm$ *STD, 9 Cluster*)

des SLRMIN-Modells sehr gut mit denen der Trainingsdaten zusammen, während die LRMIN-Modelle mit den Selbstübergängen auch bei dieser Verteilungart zu schlechteren Standardabweichungen und hier sogar abweichenden Mittelwerten führen.

Abschließend halten wir fest, dass unter den betrachteten Modellstrukturen bezüglich der relevanten Verteilungen bei den generierten Sequenzen den strikten Links-Rechts-Modellen der Vorzug gegeben werden muss. Allerdings liefern auch diese keine zufriedenstellenden SPE-Summen-Verteilungen bzw. erzeugen sehr viele unrealistische Spargeld-Sequenzen mit zu niedrigen oder zu hohen summierten Einträgen. Die Variation der Clusteranzahl hat gezeigt, dass die Kurven durch den Einsatz von mehr Modellen nicht unbedingt besser approximiert werden, sondern oft schon eine kleine Anzahl genügt.

7.5.2 HMM mit Ausgabe-Klassen (AK-HMM)

Für die folgenden Auswertungen setzen wir zunächst erweiterte Hidden-Markov-Modelle mit 6 Ausgabeklassen ein, wie wir sie in Abschnitt 7.2 beschrieben haben, und versehen diese wieder mit gestutzten Normalverteilungen (vgl. Abschnitt 7.5.1).

Die Modelle des Typs AK-HMM weisen beim Training und Generieren von Spargeld-Sequenzen im Vergleich zu den klassischen Modellen eine Besonderheit auf, die aus den bedingten Übergangswahrscheinlichkeiten in Verbindung mit stetigen Ausgabeverteilungen resultiert: Wir betrachten dazu eine Sequenz, für die bereits t Ausgaben generiert wurden und deren Zustandsfolge sich zum Zeitpunkt t in Zustand i befindet. Es ist nun möglich, dass die Ausgabe-Klasse l der bis dahin generierten Teilsequenz $O_{(t)}$ von keiner einzigen der Trainingssequenzen am Zustand i angenommen wurde, da mit den an die diskreten Daten angepassten stetigen Verteilungsfuntionen der Ausgaben auch andere Werte erzeugt werden können, als die Werte, die zum Training benutzt wurden. Die Übergangswahrscheinlichkeiten a_{ilj} dieser Ausgabe-Klasse l in jeden Zustand j wurden deshalb im vorliegenden Modell auch mit keiner Sequenz trainiert (vgl. Formel 6.9b in Abschnitt 6.3), sondern stammen aus den zufälligen Initialisierungen oder einem Zwischenschritt der Clusterung.

Im Folgenden verwenden bei den von den erweiterten Modellen generierten Sequenzen nur diejenigen, bei denen der oben beschriebene Fall nicht eingetreten ist, d. h. wir verwerfen eine Sequenz, sobald sie in einer von den Originaldaten nicht belegten Ausgabe-Klasse landet.

Modelltopologie und Clusteranzahl

Wir vergleichen wieder die generierten Sequenzen von trainierten Modellen aus jeweils zwei verschiedenen Clusterungen mit den Trainingsdaten O10K bei Verwendung der vier Modellstrukturen aus Abbildung 7.4. Somit können wir den direkten Vergleich zwischen den klassischen und den erweiterten Hidden-Markov-Modellen mit Ausgabe-Klassen ziehen.

Die Abbildungen 7.15 bis 7.18 zeigen wieder die Verteilungen der Sequenzlängen und der SPE-Summen jeweils für die Trainingsdaten O10K und die von den Modellen generierten Daten bei Verwendung von 3 und 9 Modellen bzw. Clustern.

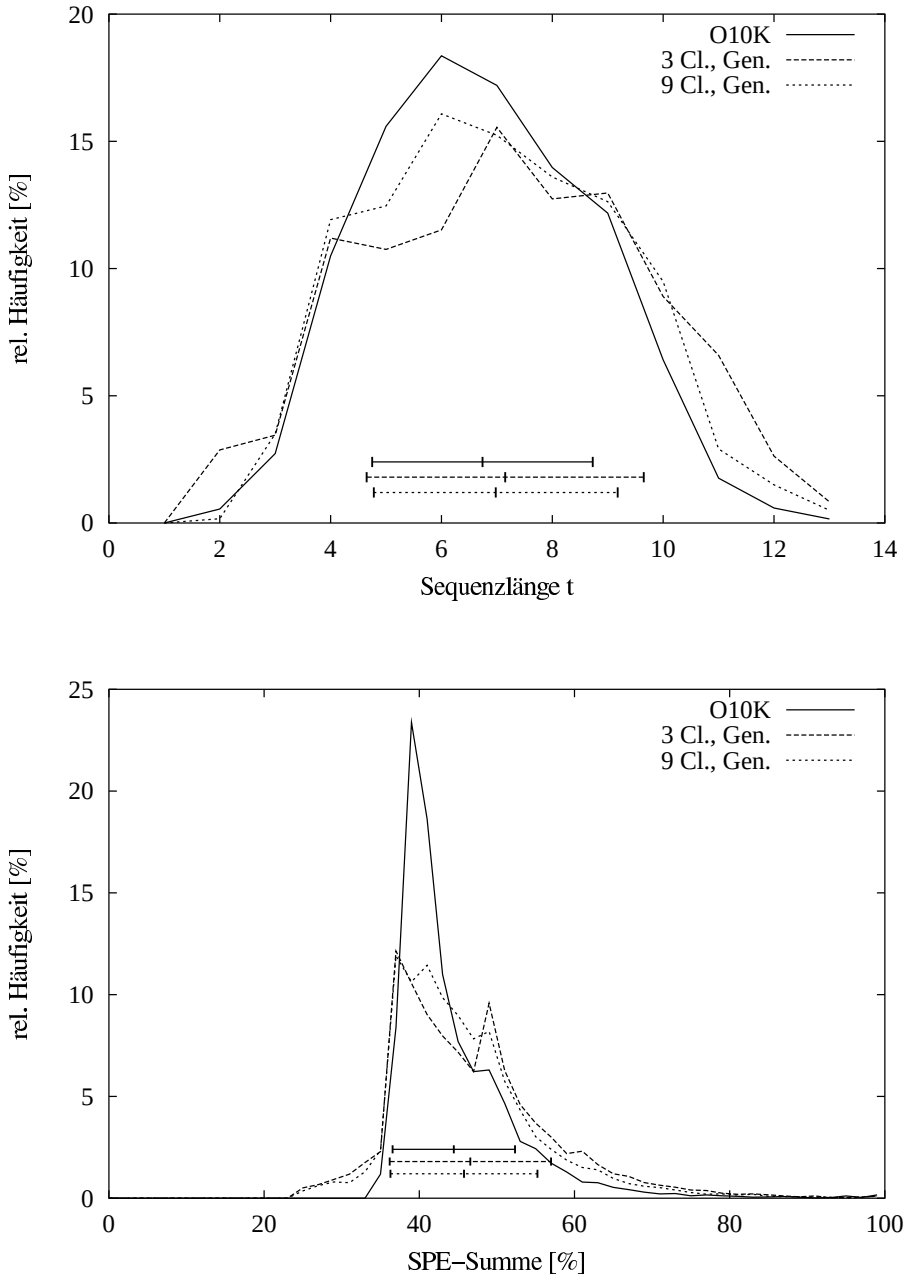


Abbildung 7.15: SLRMIN(13), AK6-HMM; Verteilungen der Trainingsdaten O10K und 10 000 generierter Daten bei verschiedenen Clusterungen (*Fehlerbalken: MW* $\pm$ *STD*)

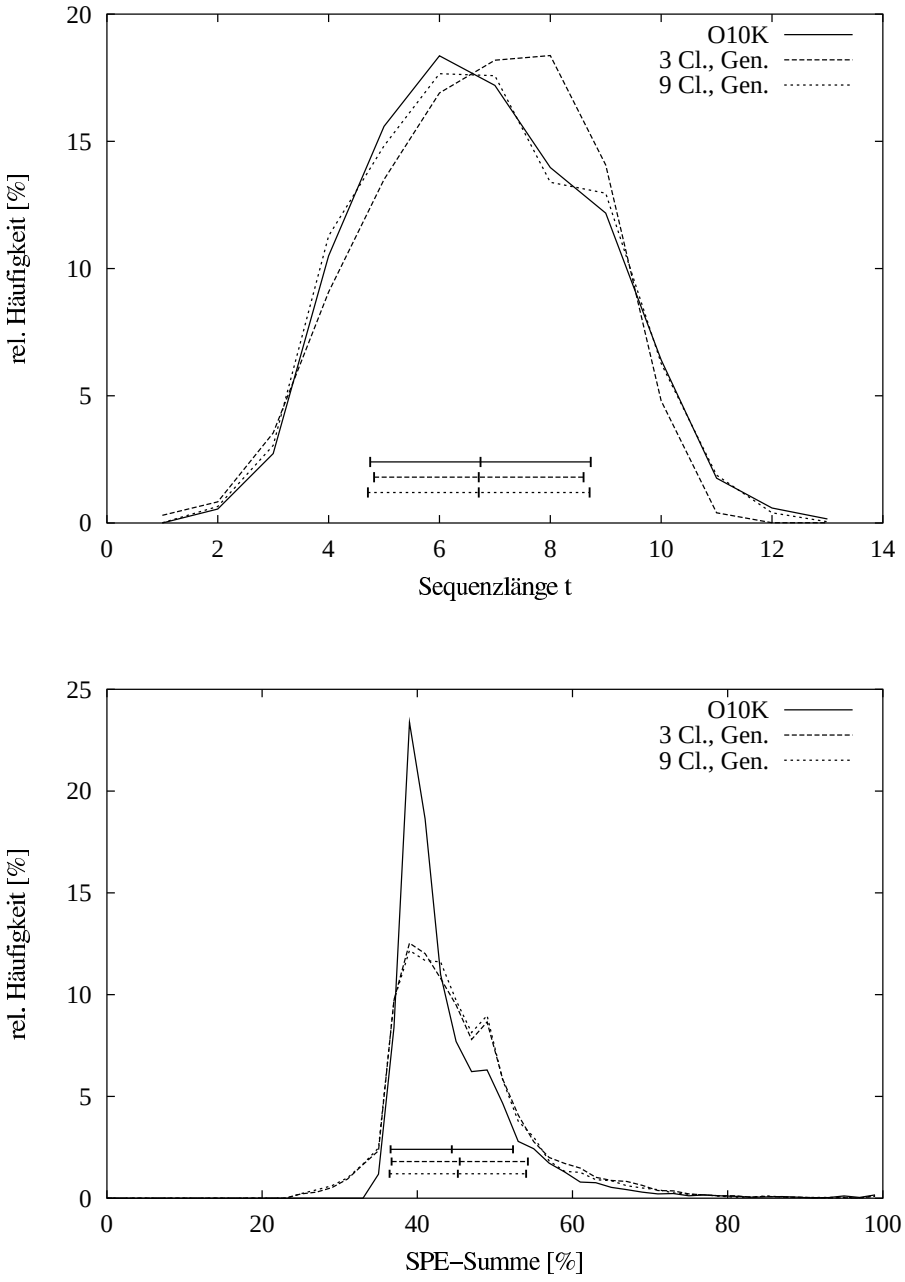


Abbildung 7.16: SLRMAX(13), AK6-HMM; Verteilungen der Trainingsdaten O10K und 10 000 generierter Daten bei verschiedenen Clusterungen (*Fehlerbalken: MW* $\pm$ *STD*)

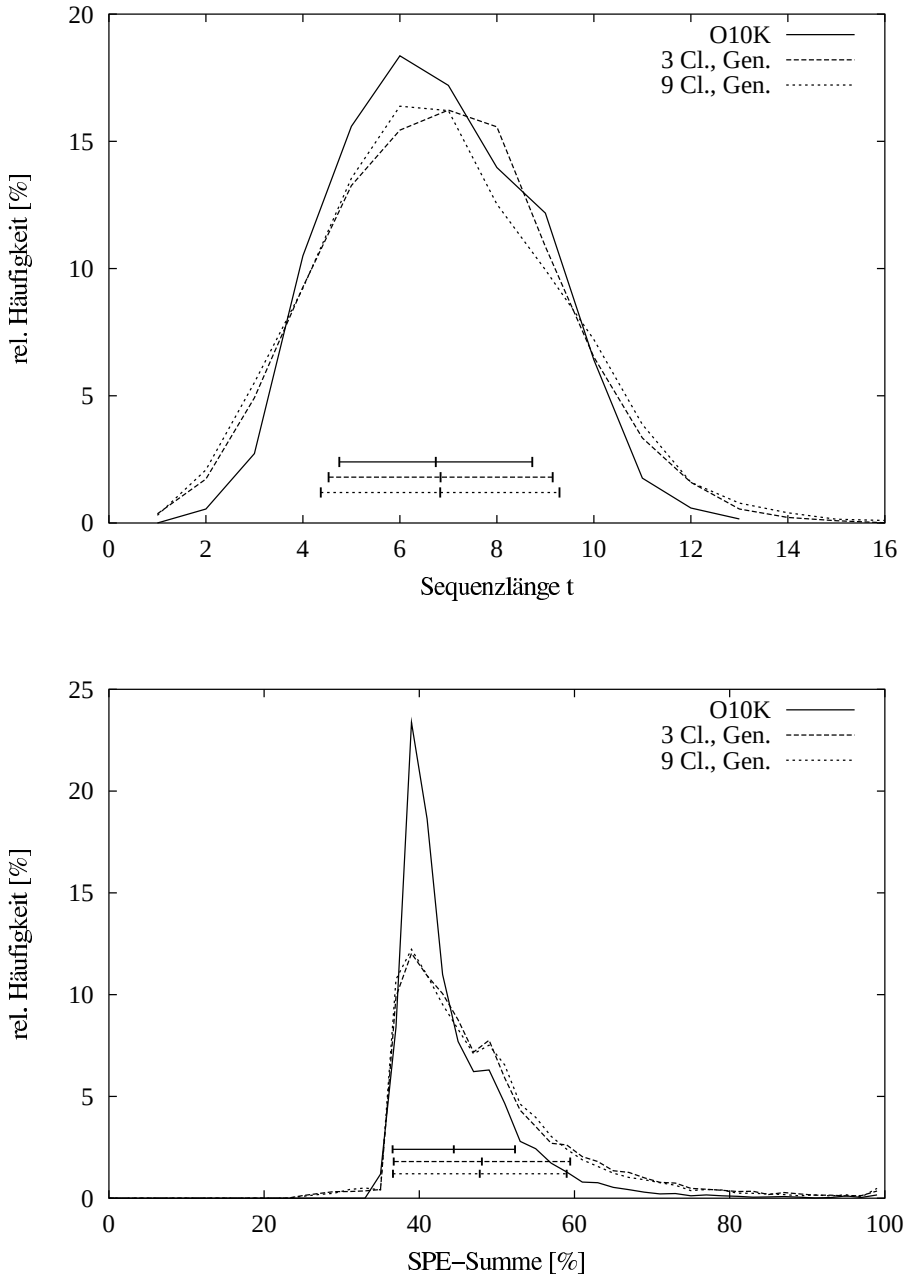


Abbildung 7.17: LRMIN(6), AK6-HMM; Verteilungen der Trainingsdaten O10K und 10 000 generierter Daten bei verschiedenen Clusterungen (*Fehlerbalken: MW* $\pm$ *STD*)

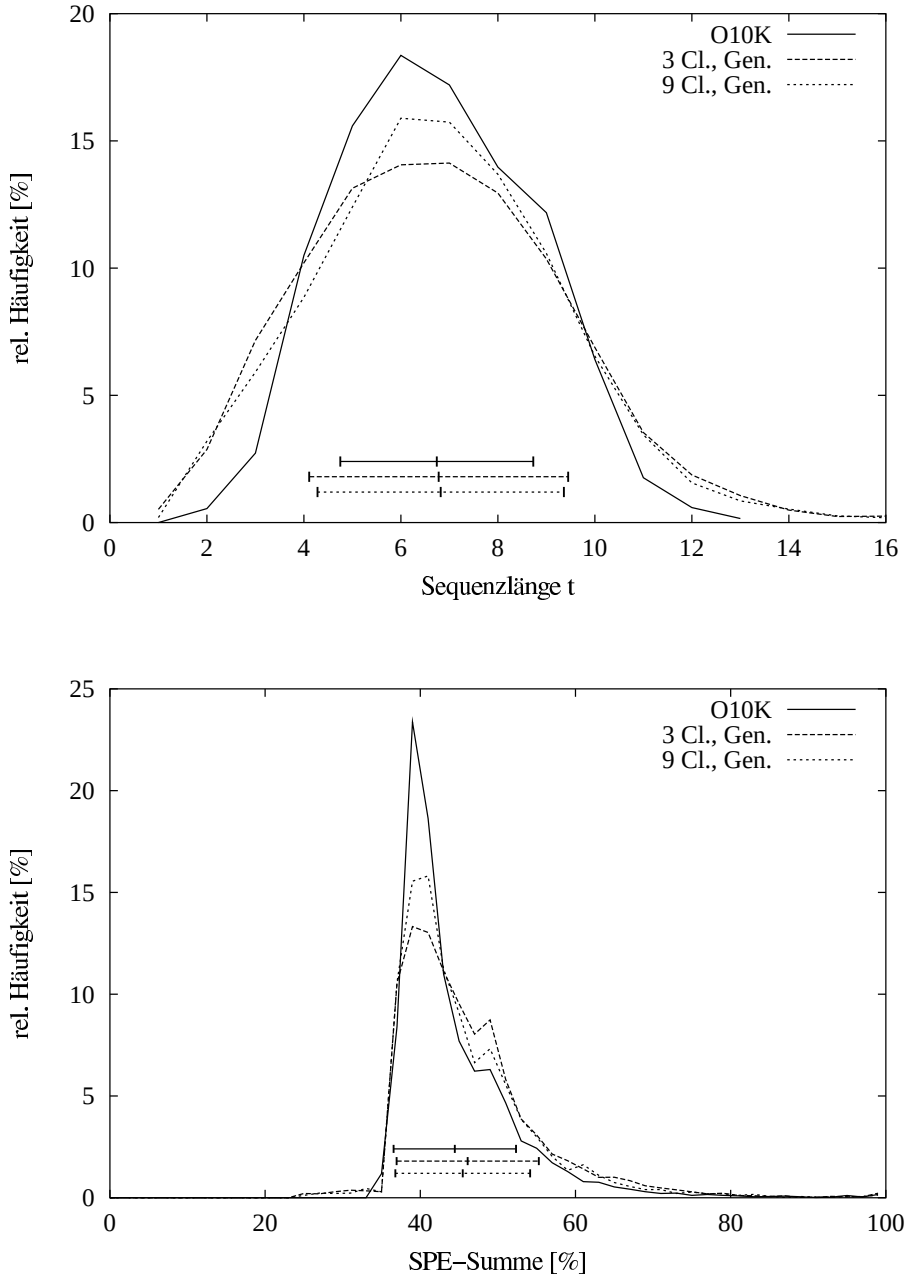


Abbildung 7.18: VG(6), AK6-HMM; Verteilungen der Trainingsdaten O10K und 10 000 generierter Daten bei verschiedenen Clusterungen (*Fehlerbalken: MW* $\pm$ *STD*)

Es zeigt sich, dass die Längenverteilung der Daten der SLRMIN(13)-Modelle schlechter geworden ist als bei Verwendung des klassischen HMM-Typs. Dies ist wahrscheinlich auf das Verwerfen einiger Sequenzen zurückzuführen, wie wir es oben beschrieben haben. Die Verteilung der SPE-Summen wurde dagegen wie erhofft durch Einsatz der Ausgabe-Klassen deutlich verbessert (vgl. Abbildungen 7.10 und 7.15). Durch Erhöhung der Clusteranzahl werden beide Verteilungen auch hier wieder etwas besser den Datenverteilungen angepasst. Auffällig ist vor allem, dass sich der Häufungspunkt bei ca. 50% SPE-Summe in den Originaldaten hier auch bei den generierten Daten widerspiegelt. Dies war bei den entsprechenden Verteilungen der mit den klassischen Modellen generierten Sequenzen praktisch nie der Fall.

Das zweite verwendete strikte Links-Rechts-Modell mit der Topologie SLRMAX(13) trifft mit seinen generierten Daten gegenüber dem SLRMIN-Modell bei der höheren Clusteranzahl $K = 9$ besser die Längenverteilung der Originaldaten O10K (Abbildung 7.16). Mit dieser Modellstruktur wird auch die Form der SPE-Summen-Verteilung der Trainingsdaten sowie deren Mittelwert und Standardabweichung besser getroffen; die Erhöhung der Clusteranzahl von 3 auf 9 verbessert die Kurve jedoch nur minimal. Erfreulich bei beiden Topologien ist die Tatsache, dass mit diesen Modellen kaum noch Sequenzen erzeugt werden, die eine im Vergleich zu den Originaldaten und den bauspartechnischen Voraussetzungen zu niedrige Summe in ihren Einträgen besitzen: Keine Sequenz weist eine kleinere SPE-Summe als 24% auf.

Die Modelltopologie LRMIN(6) mit einem Selbstübergang in jedem Zustand liefert bezüglich der relevanten Verteilungen der generierten Daten etwas schlechtere Ergebnisse als das größere SLRMAX(13)-Modell. Allerdings zeigt sich in Abbildung 7.17, dass der Einsatz der Ausgabe-Klassen des erweiterten HMM gegenüber dem KL-HMM zu einer wesentlich besseren Übereinstimmung der generierten Sequenzlängen mit denen der Trainingsdaten führt (vgl. Abbildung 7.12): Es entstehen zum einen nur wenige Sequenzen, die deutlich länger sind als die Originaldaten, so dass die Längenverteilung inklusive Standardabweichung insgesamt besser getroffen wird; zum anderen liegen die generierten Sequenzen mit ähnlicher Streuung wie der Datensatz O10K mit ihrer SPE-Summe in dem in der Realität am wahrscheinlichsten Bereich von ca. 35% bis 60%. Zudem entstehen hier nur vernachlässigbar viele Sequenzen mit kleinerer SPE-Summe als 34% (Minimum der Trainingsdaten). Ein Nachteil bei dieser Topologie ist das gegenüber den anderen Auswertungen stärkere Abweichen des Mittelwerts der SPE-Summe bei den Modellsequenzen (3 Cluster: 48.07%, 9 Cluster: 47.8%, O10K: 44.45%). Auffallend ist auch hier wieder die große Ähnlichkeit der Kurven bei 3 und 9 Clustern.

Mit der Modelltopologie VG(6) schließlich werden Sequenzen mit akzeptabler Längenverteilung und mit gegenüber allen bisherigen Auswertungen am besten passender SPE-Summen-Verteilung erzeugt (Abbildung 7.18): Bei einer Clusterung mit 9 Modellen ist eine ausgeprägte Spitze im Bereich um 40% zu sehen, und sowohl der Mittelwert als auch die Standardabweichung der Verteilungen stimmen gut überein.

Bei der Variation der Clusteranzahl K stellten wir in vielen Auswertungen fest, dass die relevanten Verteilungen der generierten Sequenzen bei wachsendem K die Verteilungen der Originaldaten nicht kontinuierlich besser approximieren, sondern oft unverändert bleiben, um sich dann bei Hinzufügen von nur einem weiteren Modell oder Cluster schlagartig zu verbessern. Dies deutet auf eine Struktur im Datensatz hin, die jeweils ab einer bestimmten Clusteranzahl von den Modellen besser abgebildet wird. Insgesamt liegt aber auch hier wieder eine große Abhängigkeit

von den Initialparametern vor.

Zusammenfassend läßt sich sagen, dass der Einsatz von Ausgabe-Klassen mittels des erweiterten HMM zu der erhofften besseren Abbildung der Nebenbedingungen der zum Modelltraining eingesetzten vollständigen Spargeld-Sequenzen führt (vgl. Abschnitt 3.3.2 sowie die Einleitung zu Kapitel 6), wobei jedoch die Längenverteilung mit den generierten Sequenzen insgesamt gesehen etwas schlechter angepasst wird. Von den hier betrachteten Modellstrukturen erscheinen uns die leider auch größeren Topologien SLRMAX(13) und VG(6) als am besten geeignet, die Spargeld-Sequenzen von zugeteilten Verträgen abzubilden, denn auch die mittleren Spargeldeingänge und die entsprechenden Standardabweichungen zeigen bei den von diesen Modellen erzeugten Daten abgesehen von den letzten drei Zeiträume, in denen aber nur noch wenige SPE-Einträge vorhanden sind, zufriedenstellende Ergebnisse (Abbildung 7.19).

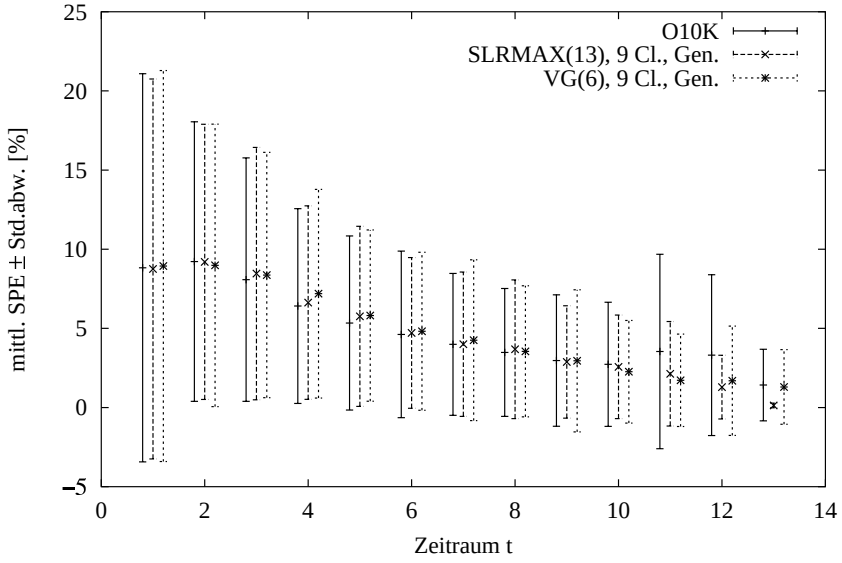


Abbildung 7.19: SLRMAX(13) und VG(6), AK6-HMM; SPE pro Zeitraum *t* der Trainingsdaten O10K und 10 000 generierter Sequenzen (*MW* $\pm$ *STD, 9 Cluster*)

Variation der Ausgabe-Klassen

Die in Abschnitt 7.2 definierten 6 Ausgabe-Klassen wurden bezüglich Anzahl und Unterteilung relativ willkürlich gewählt, wobei wir versuchten, die Anzahl der Klassen eher klein zu halten, um die statistische Unsicherheit beim Trainieren der Parameter nicht zu groß werden zu lassen. Es stellt sich die Frage, ob die relevanten Datenverteilungen mit anderen Intervallen bzw. mit weniger oder mehr Ausgabe-Klassen besser approximiert werden können.

Wir betrachten deshalb in Abildung 7.20 die diskretisierte Häufigkeitsverteilung der summierten SPE-Einträge der Sequenzen des Datensatzes O10K über alle Zeiträume *t*; d. h. im Gegensatz

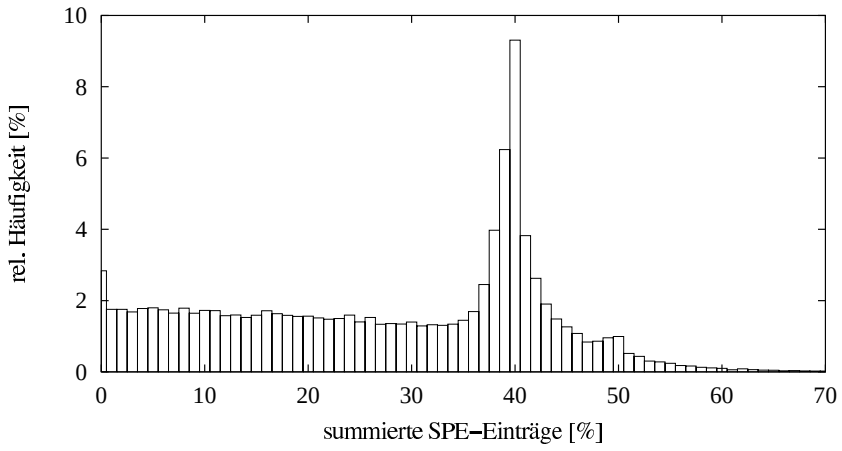


Abbildung 7.20: Verteilung der summierten SPE-Einträge der Teilsequenzen $O_{(t)}$ des Datensatzes O10K über alle Zeiträume t (*Intervallbreite* 1%)

zu den bisher betrachteten SPE-Summen der vollständigen Sequenzen gehen hier auch die SPE-Summen aller Teilsequenzen $O_{(t)} = O_1 \cdots O_t$ ein. Denn diese entscheiden beim Durchlaufen eines AK-HMM darüber, in welcher Ausgabe-Klasse der jeweilige Zustandswechsel erfolgt. Die Verteilung der Teilsequenz-Summen veranlasst uns dazu, die Intervallbreite der Ausgabe-Klassen im Bereich von 40% zu verkleinern. Wir verwenden als erste Variante ein **AK10-HMM** mit folgender Intervalleinteilung:

$$[0,5),[5,10),[10,17),[17,23),[23,30),[30,37),[37,39),[39,40),[40,44),[44,\infty)\,.$$

Die Einteilung wurde so gewählt, dass jedes Intervall ungefähr 10% aller Teilsequenz-Summen der Daten O10K enthält. Als zweite Variante betrachten wir ein einfacheres **AK3-HMM** mit den Intervallen

$$[0,35),[35,47),[47,\infty)\,.$$

Die erste rechte Intervallgrenze wird dadurch motiviert, dass die SPE-Summe aller vollständigen Sequenzen der Daten O10K mindestens 35% beträgt, die zweite durch das lokale Minimum bei 47% in Abbildung 7.20.

Mit beiden Modelltypen führten wir wieder mehrere Clusterungen pro Topologie und Clusteranzahl durch. Insgesamt unterscheiden sich die Ergebnisse nicht signifikant von den Ergebnissen mit dem AK6-Modell. In Abbildung 7.21 sind als Beispiel die SPE-Summen-Verteilungen der Topologie SLRMAX(13) für die Typen mit 3, 6 und 10 Ausgabe-Klassen aufgetragen, jeweils bei Verwendung von 9 Clustern. Die Anteile im Bereich von 40% sind zwar bei den Modellen mit den obigen Intervalleinteilungen etwas höher als bei dem AK6-Modell; Mittelwerte und Standardabweichungen sind jedoch vergleichbar, wobei hier das AK3-HMM geringfügig schlechter abschneidet.

In Abbildung 7.22 sind die gleichen Kurven für die Topologie VG(6) zu sehen. Hier erstaunt uns die Tatsache, dass die Verteilung der Trainingsdaten mit den Sequenzen der einfacheren AK3-

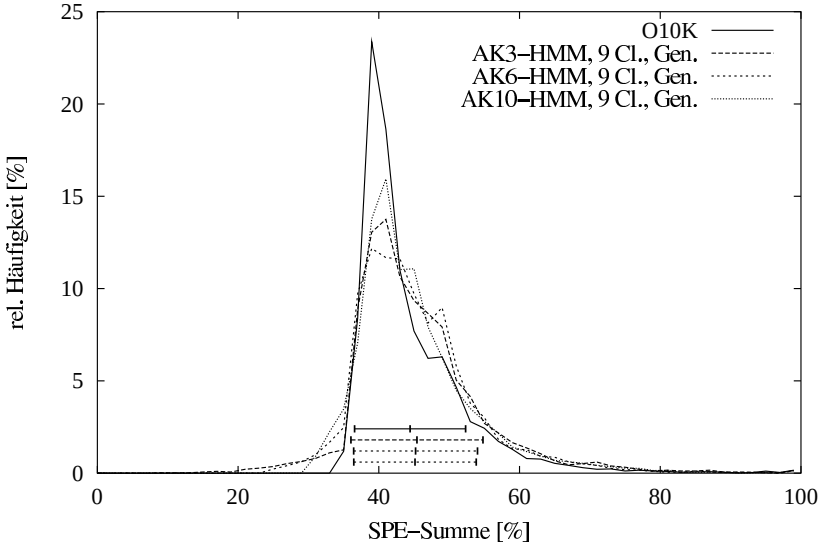


Abbildung 7.21: SLRMAX(13), AK-HMM mit 3, 6 und 10 Ausgabe-Klassen; SPE-Summen der Trainingsdaten O10K und 10 000 generierter Daten ($K = 9$, *Fehlerbalken: MW* $\pm$ *STD*)

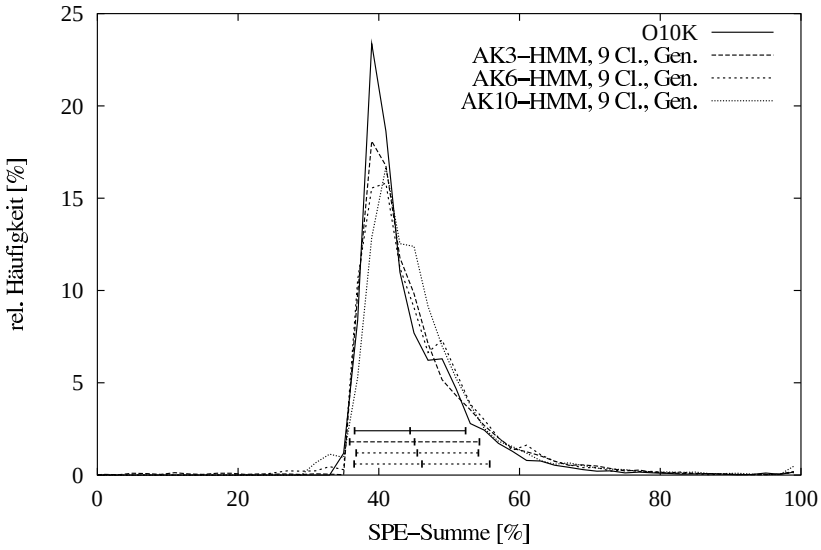


Abbildung 7.22: VG(6), AK-HMM mit 3, 6 und 10 Ausgabe-Klassen; SPE-Summen der Trainingsdaten O10K und 10 000 generierter Daten ($K = 9$, *Fehlerbalken: MW* $\pm$ *STD*)

Modelle besser approximiert wird, während mit den AK10-Modellen der Mittelwert am stärksten abweicht.

Die Längenverteilungen aller mit den beiden Modelltypen mit 3 und 10 Ausgabe-Klassen erzeugten Sequenzen änderten sich im Vergleich zum AK6-HMM ebenfalls kaum. Insgesamt

konnten wir feststellen, dass mit den 3 Ausgabe-Klassen die Datenverteilungen der Originaldaten im Schnitt genauso gut approximiert werden können wie mit den beiden anderen Modelltypen; allerdings scheint die Abhängigkeit von Clusteranzahl und Initialisierung bei dem einfacheren Modelltypen größer zu sein. Wie schon bei dem AK6-HMM zeigt sich auch bei den hier betrachteten Typen, dass die Verteilungskurven mit steigender Clusteranzahl nicht unbedingt besser werden.

Abschließend lässt sich sagen, dass unsere ursprüngliche Einteilung von fünf äquidistanten Intervallen plus einem „Rest"-Intervall (siehe Abschnitt 7.2) letztendlich eine gute Wahl ist, da mit dem AK6-HMM „stabilere" Ergebnisse als mit dem AK3-HMM erzielt werden; mit dem AK10-HMM dagegen lassen sich die Verteilungskurven auch nicht besser approximieren. Bei größerer Anzahl von Ausgabe-Klassen erhöht sich zudem die statistische Unsicherheit beim Trainieren der Modellparameter, und die Übergangswahrscheinlichkeiten werden in vielen Ausgabe-Klassen auf null gesetzt.

7.5.3 Vergleich mit dem *K*-means-Verfahren

In dem zur Zeit in der Praxis eingesetzten mesoskopischen Simulationsmodell für Bausparkollektive werden mittels des *K*-means-Verfahrens aus vollständigen Spargeld-Sequenzen Prototypen gewonnen (siehe Abschnitt 3.2.3). Dabei werden für jeweils eine Tarifklasse zwischen 20 und 30 Prototypen bestimmt, die in einer Simulation das Sparverhalten der einzelnen Bausparer repräsentieren. Es bietet sich deshalb ein ähnlicher Vergleich der *K*-means-Prototypen mit den Trainingsdaten an, wie wir ihn für die generierten Sequenzen vorgenommen haben.

Eine Clusterung der Daten O10K mit dem *K*-means-Verfahren, wie es im Abschnitt 3.2.3 beschrieben wurde, führt zu einer Partitionierung der Daten in K Cluster mit jeweils einem Zentralpunkt, der das arithmetische Mittel der Sequenzen eines Clusters darstellt. Da die unterschiedlich langen Trainingssequenzen für das Verfahren mit Nullen aufgefüllt werden, weisen alle Zentralpunkte T_{max} Einträge auf. Als Prototyp eines Clusters übernehmen wir die Zentralpunkte, kürzen sie aber auf die mittlere Länge der Daten des entsprechenden Clusters. Jeder Prototyp erhält zusäzlich eine Gewichtung zugewiesen, die dem relativen Anteil der Sequenzen seines Clusters an allen Trainingssequenzen entspricht.

In Abbildung 7.23 sehen wir die Verteilungen der gewichteten Prototypen. Dazu wurden deren nicht ganzahligen Längen zur Vergleichbarkeit mit der Längenverteilung der Trainingsdaten in die Intervalle

$$[0.5, 1.5), [1.5, 2.5), \ldots, [12.5, 13.5)$$

einsortiert und die SPE-Summen entsprechend in die für die Originalsequenzen ebenso verwendeten Intervalle der Breite 2%. Die Anteile der Prototypen entsprechen damit den aufsummierten Gewichten der jeweiligen Intervallklasse.

Nach Konstruktion stimmen die Mittelwerte der Längen überein, während sich bei den Prototypen eine etwas kleinere Streuung als bei den Trainingsdaten ergibt. Die Form der Verteilung selbst wird einigermaßen gut getroffen, wobei sich durch die Mittelung der Daten zu wenigen

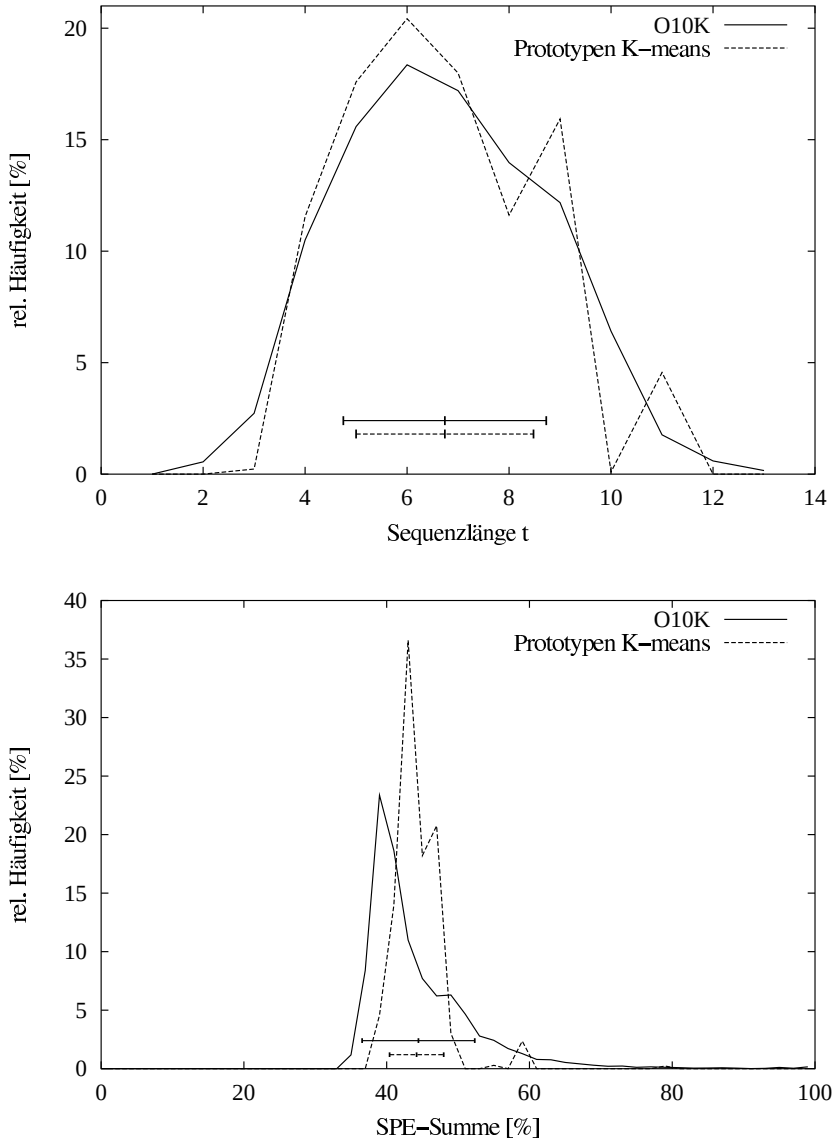


Abbildung 7.23: Verteilungen der Trainingsdaten O10K und der 25 gewichteten Prototypen einer *K*-means-Clusterung (*Fehlerbalken: MW* $\pm$ *STD*)

Repräsentanten mehrere lokale Minima und Maxima in der Verteilungskurve bilden. Auffällig bei den SPE-Summen ist die Verschiebung der maximalen Häufigkeit auf der *x*-Achse nach rechts. Die summierten Einträge der Prototypen konzentrieren sich größtenteils um die Werte von 43% und 47%, und dadurch ergibt sich auch eine viel kleinere Standardabweichung als in den Originaldaten.

Im Vergleich zu den Häufigkeitsverteilungen der von den erweiterten Hidden-Markov-Modellen generierten Sequenzen schneiden die Prototypen des *K*-means-Verfahrens insgesamt eher schlechter ab. Es ist jedoch schwierig, eine konkrete qualitative Aussage zu treffen, da die Modellierungsansätze bei beiden Verfahren sehr unterschiedlich sind.

7.6 Struktur einer Clusterung

Die bisherigen Auswertungen bezogen sich weitgehend auf die Gesamtheit der Modelle bzw. der genierten Sequenzen einer Clusterung. Wir wollen deshalb an dieser Stelle kurz auf einzelne Cluster eingehen, die bei dem Maximum-Likelihood-Verfahren entstehen.

Clusteranteile

Verschiedene Clusterverfahren führen i. d. R. zu unterschiedlichen Clustergrößen. So bildet z. B. das *K*-means-Verfahren relativ gleich große Gruppen, während der ebenfalls geometrische Single-Link-Algorithmus [18,44] dazu neigt, Ausreißer in den Daten in eigene Cluster zu setzen und daneben einige wenige große Gruppen zu bilden. Zur Simulation des Sparer-Kollektivs sollten jedoch sehr kleine Cluster, deren Modelle das Sparverhalten von wenigen, nicht repräsentativen Sparern abbilden, vermieden werden. Erfreulicherweise stellt sich heraus, dass die Sequenzen bei dem Maximum-Likelihood-Verfahren recht gleichmäßig auf die Modelle verteilt werden.

In Tabelle 7.2 sind beispielhaft die relativen Clustergrößen für die Topologien SLRMIN(13) und VG(6) bei variierender Clusteranzahl K zu sehen. Es fällt auf, dass die Anteile bei Verwendung der flexibleren Modellstruktur VG(6) für eine größere Clusteranzahl stärker schwanken als die Anteile bei Verwendung der Struktur SLRMIN(13). Bei beiden Topologien entstehen jedoch keine extrem kleinen oder großen Cluster.

Cluster-Nr.:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Topologie, Clusteranz.	*Anteil der Sequenzen des Datensatzes O10K* [%]											
SLRMIN(13), $K = 4$	19.1	32.0	21.9	27.0								
VG(6), $K = 4$	17.6	30.2	33.9	18.3								
SLRMIN(13), $K = 8$	15.5	10.4	10.6	12.0	13.4	13.0	14.5	10.6				
VG(6), $K = 8$	15.8	11.3	13.7	26.3	8.0	5.1	18.6	1.2				
SLRMIN(13), $K = 12$	10.3	3.9	8.1	9.1	6.3	9.5	10.5	11.9	9.1	6.3	7.8	7.2
VG(6), $K = 12$	5.0	4.2	1.7	6.6	8.0	15.0	25.8	7.9	10.1	6.7	2.2	7.6

Tabelle 7.2: Anteile der Sequenzen pro Cluster (AK6-HMM, Datensatz O10K)

SPE-Verteilung der Cluster

Das Ausgangsziel unserer Modellbildung und der Clusterung der Daten mit mehreren Modellen war unter anderem das Trennen der Trainingsdaten in Gruppen mit typischem und ähnlichem Sparverhalten. In den folgenden Auswertungen sollen deshalb für einzelne Beispiele die Mittelwerte und Varianzen der Spargeldeingänge pro Zeitraum t innerhalb eines Clusters der Trainingsdaten betrachtet werden. Im Gegensatz zu den Abbildungen 7.2, 7.14 und 7.19, bei denen wir den Mittelwert und die Standardabweichung pro Zeitraum t über die in t vorhandenen SPE-Einträge gebildet hatten (vgl. Abschnitt 7.1), addieren wir hier die Spargeldeingänge eines Zeitraums und teilen die Summe durch die Anzahl *aller* Sequenzen des Clusters; damit erhalten wir die mittleren Sparbeiträge der Verträge eines Clusters im Zeitraum t, die dem Bausparkollektiv zufließen.

Wir betrachten zwei Clusterungen mit jeweils 15 Modellen, einmal vom Typ SLRMIN(13) und einmal vom Typ VG(6). Abbildung 7.24 zeigt, dass bei der Clusterung Gruppen entstehen, die sich deutlich voneinander abgrenzen. Die Verläufe im ersten Bild entsprechen sogenannten Soforteinzahlern, die kurz nach Vertragsabschluss bereits das zur Zuteilung benötigte Guthaben eingezahlt haben. Das Cluster des SLRMIN-Modells weist allerdings eine wesentlich geringere Streuung auf als das Cluster der VG-Topologie. Während bei ersterem fast ausschließlich die „40%-Einzahler“ des ersten Jahres zusammengefasst sind, bildet das VG-Cluster eine Mischung aus Soforteinzahlern im ersten und im zweiten Jahr.

Im zweiten Bild dagegen sind zwei Cluster von sogenannten Regelsparern zu sehen. Das Cluster

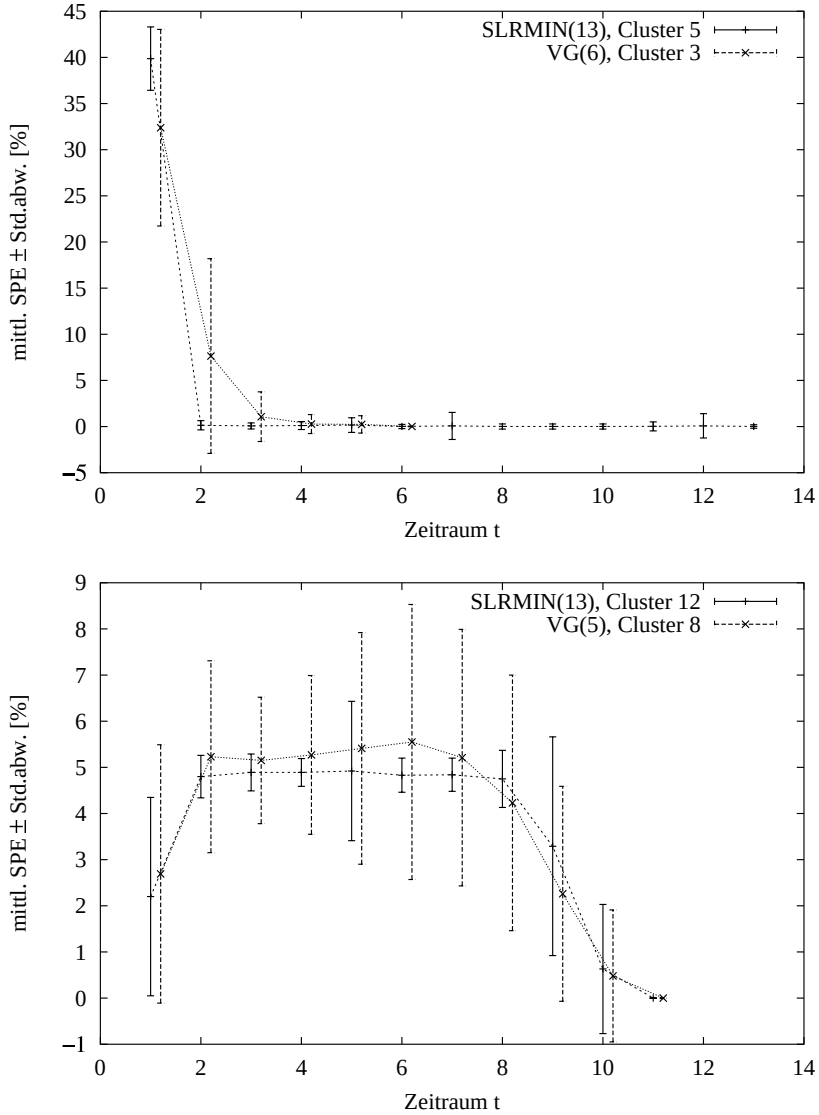


Abbildung 7.24: SPE pro Zeitraum *t* von Trainingsdaten ausgewählter Cluster (*MW ± STD, Clustermodelle SLRMIN(13) und VG(6), AK6-HMM, 15 Cluster*)

der SLRMIN-Topologie zeigt die typische Einzahlungskurve, die der Regelsparrate des Tarifs von 4.8% im Jahr entspricht. Auch hier werden im VG-Cluster mehrere verschiedene Sparformen zusammengefasst, was sich in der größeren Varianz ausdrückt.

Die hier gezeigten Kurven stellen Beispiele von Sparergruppen dar, deren Sparverlauf in ein bestimmtes Schema passt und bei den Bausparkassen als typisch bekannt ist; im Allgemeinen entstehen jedoch viele Cluster, bei denen die Sequenzen sehr inhomogenen Sparverläufen entsprechen, d. h. Cluster mit großen Streuungen und jährlich stark schwankenden Mittelwerten im SPE. Wie an den obigen Kurven schon andeutungsweise zu sehen ist, werden bei einer Clusterung mit der SLRMIN-Topologie i. d. R. Gruppen mit homogeneren Sparverläufen – sowohl pro Zeitraum als auch über die Jahre hinweg – gebildet, als bei Einsatz der anderen Topologien.

Im vorigen Abschnitt hatten wir gesehen, dass sich eine Erhöhung der Cluster- und Modellanzahl teilweise nur gering auf die Gesamt-Verteilungen der generierten Sequenzen auswirkt. Bei der Begutachtung der einzelnen Cluster der Trainingsdaten stellt sich jedoch heraus, dass die Sequenzen innerhalb eines Clusters bei wachsender Anzahl von Modellen besser zusammenpassen und sich homogenere Gruppen bilden. Die Hidden-Markov-Modelle, die solche Gruppen mit typischem und ähnlichem Sparverhalten abbilden, sind in einer Simulation besser zu handhaben.

Die Ergebnisse dieses Abschnittes sprechen dafür, bei der Wahl der Clusteranzahl K neben den Verteilungen der generierten Sequenzen auch die Clustergrößen und die Verteilungen innerhalb der gruppierten Trainingsdaten zu berücksichtigen, damit einerseits nicht zu kleine Cluster entstehen und andererseits die Sequenzen in relativ homogenen Gruppen zusammengefasst werden.

7.7 Laufzeiten

Bevor wir das Kapitel mit einer Diskussion der Ergebnisse beschließen, vergleichen wir die Laufzeiten des Maximum-Likelihood-Verfahrens zur Clusterung der 10 000 Trainingssequenzen O10K bei Verwendung der verschiedenen Topologien, Dichtefunktionen und Modelltypen. In Tabelle 7.3 sind die mittleren Laufzeiten aus je drei Läufen mit zufälligen Startpartitionen eingetragen; alle Clusterungen wurden mit Hilfe unserer HMM-Bibliothek (Programmiersprache C, gcc-Compiler, Betriebssystem UNIX) auf einer SUN ET 450 gerechnet, die mit UltraSPARC-II-Prozessoren (400 MHz) ausgestattet ist.

Die Rechenzeiten hängen von den Initalisierungen der Clusterung und der daraus resultierenden Anzahl von Iterationen ab, so dass die individuellen Zeiten stark schwanken. Generell können wir aber feststellen, dass die Modelltypen des AK-HMM gegenüber dem klassischen Modelltyp kaum längere CPU-Zeiten benötigen. Auch die Erhöhung der Ausgabe-Klassen wirkt sich nur wenig auf die Laufzeiten aus, während die Verdoppelung der Clusteranzahl die Werte bei beiden HMM-Typen signifikant ansteigen lässt. Die kleinsten Laufzeiten werden fast immer mit der Normalverteilung erreicht; dies liegt daran, dass bei bei der Reestimierung der Parameter für die gestutzte Normalverteilung sehr oft auf die tabellierten Φ-Werte zugegriffen werden muss und eine numerische Nullstellensuche erforderlich ist (vgl. Abschnitte 5.2.3 und 5.2.5).

Die kürzeren Laufzeiten der SLRMIN-Topologie gegenüber der kleineren LRMIN-Topologie, die weniger Knoten und Kanten aufweist (vgl. Tabelle 7.1 in Abschnitt 7.2), lassen sich durch

Modelltyp	*Cluster*	*Dichte-Fkt.*	*SLRMIN(13)*	*SLRMAX(13)*	*LRMIN(6)*	*VG(6)*
AK3-HMM	10	gest. NV	12:38	50:33	21:27	33:46
AK6-HMM	10	gest. NV	15:24	56:58	22:19	34:01
AK10-HMM	10	gest. NV	26:11	61:38	24:52	32:02
AK6-HMM	20	gest. NV	32:11	76:48	35:08	39:56
AK6-HMM	10	NV	11:23	51:18	20:38	30:41
KL-HMM	10	gest. NV	25:24	46:52	24:31	28:31
KL-HMM	20	gest. NV	32:49	69:06	32:12	52:22
KL-HMM	10	NV	13:55	39:59	15:29	26:43

Tabelle 7.3: Laufzeiten verschiedener Clusterungen [m:s] (*CPU: UltraSPARC-II, 400 MHz; MW aus je 3 Zufalls-Initialisierungen; Daten O10K*)

programmiertechnische Vereinfachungen in den Berechnungen erklären, die bei den einfachen SLRMIN-Strukturen, bei denen jedes Sequenzsymbol einem eindeutigen Zustand zugeordnet werden kann, verstärkt greifen.

Gegenüber den relativ langen Rechenzeiten für die Clusterungen werden zum Generieren von 10 000 Sequenzen auf der gleichen Maschine nur wenige Sekunden benötigt.

7.8 Diskussion der Ergebnisse

Die Anwendungsbeispiele der vorigen Abschnitte stellen nur eine beschränkte Auswahl unter der Vielzahl der möglichen Untersuchungen dar. Wir haben dabei unser Hauptaugenmerk auf die in dieser Arbeit theoretisch behandelten Methoden und Erweiterungen gerichtet. Abschließend fassen wir die wichtigsten Erkenntnisse zusammen:

Von den vorgeschlagenen Initialisierungen für das Maximum-Likelihood-Verfahren erwies sich eine Zufallspartition der Sequenzen als geeigneter als die Zuordnung der Sequenzen zu zufällig erzeugten Modellen (Abschnitt 7.3). Bei allen Auswertungen zeigte sich jedoch insgesamt eine starke Abhängigkeit der Clusterung von den Startwerten. Da das Maximum-Likelihood-Verfahren sowohl bei jedem Modelltraining als auch durch den Sequenzwechsel zum jeweils besten Modell in jeder Iteration nur zu einem lokalen Maximum der Zielfunktion führt, ist dies nicht erstaunlich; das Verfahren sollte jedoch in diesem Punkt noch verbessert werden, z. B. durch eine geeignete Modifikation der trainierten Parameter mit anschließendem Neutraining, um auf diese Weise ein lokales Maximum zu überwinden.

Die in Kapitel 4 vorgestellten Indizes erwiesen sich bei unseren Daten und Modellen mehr oder weniger als unzureichend für die Bewertung einer Clusterung; einzig der ID-Index konnte einen schwachen Hinweise auf eine geeignete Clusteranzahl oder eine geeignete Anzahl von Zuständen in einem HMM geben (Abschnitt 7.4). Bezüglich der trainierten Modelle besteht das Problem, dass kein geeignetes Abstandsmaß vorliegt: Das in [22] vorgeschlagene Maß $D_s(\lambda_1, \lambda_2)$

zwischen den Modellen λ_1 und λ_2 liefert den Wert ∞ zurück, sobald eines der beiden Modelle eine Sequenz erzeugen kann, die vom anderen Modell nicht abgebildet wird. Hier könnte die Übertragung des in [30] beschriebenen Abstands für Modelle mit diskreten Ausgaben auf stetige Ausgabeverteilungen eine Alternative darstellen. Eine weitere Möglichkeit liegt darin, bei der Bestimmung des Abstands $D_s(\lambda_1, \lambda_2)$ zunächst nur Sequenzen heranzuziehen, die von beiden Modellen erzeugt werden können und evtl. sonstige generierte Sequenzen über ein geeignetes „Strafmaß“ gesondert zu berücksichtigen.

In Abschnitt 7.5 haben wir uns ausführlich mit dem Generieren von Sequenzen mittels der bei einer Clusterung trainierten Modelle beschäftigt und die Verteilungen dieser Sequenzen mit denen der Trainingsdaten verglichen. Es hat sich gezeigt, dass die Verwendung von gestutzten Normalverteilungen zu deutlich besserer Approximation des Spargeldeingangs führt. Ein Vergleich der Verteilungen von generierten Sequenzen bei variierender Modelltopologie und Clusteranzahl machte deutlich, dass das um Ausgabe-Klassen erweiterte HMM die deterministische Nebenbedingung der Zuteilung, die sich in den SPE-Summen niederschlägt, im Gegensatz zum klassischen Modell wesentlich besser abbilden kann. Durch Variation der Ausgabe-Klassen stellten wir allerdings auch fest, dass es schwierig ist, eine „optimale“ Klasseneinteilung anzugeben. Verglichen mit den entsprechend aufgestellten Verteilungen der deterministischen Prototypen einer *K*-means-Clusterung liefern die Sequenzen der erweiterten Hidden-Markov-Modelle insgesamt jedoch eine zufriedenstellende Approximation der Originaldaten.

In Abschnitt 7.6 konnten wir sehen, dass das HMM-basierte Clusterverfahren die Trainingssequenzen je nach verwendeter Topologie in Gruppen vergleichbarer Größe zusammenfasst, die zum Teil ein typisches und in der Realität bekanntes Sparverhalten repräsentieren. Im letzten Abschnitt schließlich stellten wir die Laufzeiten des Clusterverfahrens für verschiedene HMM-Architekturen gegenüber. Dabei zeigte sich, dass sich von den Erweiterungen die gestutzte Normalverteilung stärker auf die Rechenzeit auswirkt als der Einsatz von Modellen mit Ausgabe-Klassen.

Ein Problem der Hidden-Markov-Modellierung stellt sicherlich die Vielzahl der unbekannten Parameter dar, für die es größtenteils noch keine Bewertungskriterien gibt und die den jeweiligen Anwendungen angepasst werden müssen (Anzahl der Zustände, Topologie, Initialparameter, Clusteranzahl). Die Ausgabe-Klassen unseres erweiterten HMM vergrößern dabei zusätzlich die Anzahl der freien Parameter der Modelle. Dennoch zeigen die vorliegenden Ergebnisse unserer Meinung nach, dass eine Modellierung der Zeitreihen eines Bausparkollektivs mit Hidden-Markov-Modellen erfolgreich umzusetzen ist, eine gute Alternative zu der deterministischen Abbildung über die Prototypen einer *K*-means-Clusterung darstellt und somit die Basis für die in Abschnitt 3.3.1 formulierte Idee eines HMM-basierten Simulationsmodells bilden kann.

Kapitel 8

Zusammenfassung und Ausblick

In dieser Arbeit haben wir einen auf Hidden-Markov-Modellen (HMM) basierenden Modellierungsansatz für ökonomische Zeitreihen, die ein Bausparkollektiv beschreiben, vorgeschlagen. Mit diesem Modellierungsansatz konnten wir Spargeld-Zeitreihen der Verträge einer Bausparkasse analysieren und simulieren. Zur Abbildung spezieller Nebenbedingungen dieser Zeitreihen wurden Erweiterungen der HMM-Theorie notwendig.

Ein HMM ist ein stochastisches Modell zur Beschreibung von Zeitreihen bzw. Sequenzen, das diese mit einer bestimmten Wahrscheinlichkeit produzieren kann. Es kann mit Hilfe bekannter Algorithmen so trainiert werden, dass die Wahrscheinlichkeit oder Likelihood für eine gegebene Menge von Sequenzen ein lokales Maximum annimmt.

In Anlehnung an bekannte Algorithmen konnten wir zunächst ein HMM-basiertes Clusterverfahren aufstellen, das eine Menge von Sequenzen in eine vorgegebene Anzahl von Gruppen unterteilt und bei dem gleichzeitig für jede Gruppe ein HMM trainiert wird, so dass die Gesamt-Likelihood lokal maximiert wird. Zur Bewertung einer Clusterung haben wir verschiedene Kenngrößen entwickelt, die bei kleinen Clusterproblemen mit strukturierten Daten gute Resultate erbringen, im Zusammenhang mit weniger strukturierten Daten und komplexeren Modellen jedoch nur eingeschränkte Aussagen zulassen.

Die sich anschließenden theoretischen Erweiterungen für Hidden-Markov-Modelle wurden durch die Spargeld-Zeitreihen eines Bausparkollektivs motiviert. Zum einen konnten wir zeigen, dass sich als Verteilungsfunktion für die Ausgaben eines HMM auch eine gestutzte Normalverteilung zur Abbildung von nichtnegativen Sequenzen einsetzen lässt. Dazu haben wir die Reestimierungsgleichungen der entsprechenden Parameter aufgestellt und nachgewiesen, dass diese sich numerisch lösen lassen, wobei sich die Laufzeit des Modelltrainings nur um einen konstanten Faktor erhöht. Die vertraglichen Nebenbedingungen der Spargeldeingänge einer Bausparkasse haben andererseits zur Definition eines HMM mit Ausgabe-Klassen geführt, bei dem die weitere Entwicklung des internen stochastischen Prozesses zum Zeitpunkt t auch von der bis dahin modellierten bzw. erzeugten Teilsequenz abhängt. Wir konnten die Basisalgorithmen für Hidden-Markov-Modelle auf das erweiterte Modell anpassen und haben gezeigt, dass deren Komplexität sich gegenüber dem ursprünglichen Modell nicht ändert, wenn die Zahl der Ausgabe-Klassen kleiner ist als die maximale Sequenzlänge.

In dem Anwendungsteil der Arbeit haben wir das HMM-basierte Clusterverfahren zur Partitionierung und Modellierung eines Realdatensatzes von Spargeldeingängen einzelner Bausparverträge eingesetzt. Unter Variation der HMM-Architektur wurde neben dem Einfluss verschiedener Initialisierungen auf die Zielfunktion und der Praxistauglichkeit der Clusterindizes die Sequenzgenerierung mit den trainierten Modellen eingehend analysiert. Dabei hat sich herausgestellt, dass die vorgeschlagenen HMM-Erweiterungen zu deutlich besseren Ergebnissen führen, besonders bezüglich der Übereinstimmung der Spargeldeingänge und deren Summen von generierten Sequenzen und Trainingsdaten. Verglichen mit deterministischen Prototypen, wie sie im derzeit in der Praxis eingesetzten Simulationsmodell verwendet werden, konnten wir die Trainingsdaten durch generierte Sequenzen insgesamt besser approximieren.

Die Ergebnisse dieser Arbeit lassen den Schluss zu, dass die in anderen Bereichen bereits erfolgreich eingesetzten Hidden-Markov-Modelle bei entsprechenden Modifikationen und Erweiterungen dazu geeignet sind, die Zeitreihen eines Bausparkollektivs und deren speziellen Nebenbedingungen abzubilden. Somit können sie als Grundlage für ein neues Simulationsmodell dienen, bei dem der Ablauf eines kompletten Bausparvertrags über ein HMM modelliert wird und von dem zu erwarten ist, dass es die Varianzen in den Daten besser erfassen kann als das zur Zeit in der Praxis eingesetzte mesoskopische Modell. Ein solches Gesamt-HMM ist bereits Gegenstand aktueller Untersuchungen der Arbeitsgruppe am ZAIK. In diesem Rahmen wird zu klären sein, inwieweit sämliche Aktionsmöglichkeiten eines Sparers über ein klassisches oder über ein erweitertes HMM mit Ausgabe-Klassen abgebildet werden können. Neben dem Einsatz von weiteren Modell-Modifikationen, wie sie in anderen Anwendungsgebieten zum Teil eingesetzt werden, ist auch eine nachgeschaltete Verarbeitung von generierten Sequenzen denkbar, etwa vergleichbar mit der Kontextanalyse bei der automatischen Spracherkennung.

Daneben bieten sowohl das von uns aufgestellte Clusterverfahren als auch das erweiterte HMM Raum für künftige theoretische Untersuchungen. Ein Ziel sollte sein, die starke Abhängigkeit der Clusterung von den Startwerten einzuschränken bzw. Möglichkeiten zu finden, ein lokalen Maximum wieder zu verlassen, um die Zielfunktion weiter zu verbessern. Eine andere wertvolle Hilfe für die Bewertung von trainierten Modellen würde ein verbessertes Abstandsmaß darstellen, das auch Modelle sinnvoll vergleichen kann, die zum Teil unterschiedliche Sequenzen generieren. Schließlich liegt es nahe, die diskreten Ausgabe-Klassen des erweiterten HMM und deren ebenfalls diskreten Übergangswahrscheinlichkeiten durch stetige Verteilungsfunktionen zu ersetzen, deren Argumente die bis dahin ausgegebene Teilsequenz bzw. im speziellen Fall die Summe dieser Ausgaben sind. Neben der Schaffung einer fundierten theoretischen Basis wäre es für ein solches Modell auch interessant, die Gemeinsamkeiten und Unterschiede zu anderen generalisierten Hidden-Markov-Modellen aufzuzeigen.

Anhang A

Reestimierungsformeln

HMM mit diskreten Ausgaben

$$\overline{\pi}_i = \frac{\sum_{k=1}^{K} \hat{\alpha}_1^k(i)\,\hat{\beta}_1^k(i)}{\sum_{k=1}^{K}\sum_{j=1}^{N} \hat{\alpha}_1^k(j)\,\hat{\beta}_1^k(j)},$$

$$\overline{a}_{ij} = \frac{\sum_{k=1}^{K}\sum_{t=1}^{T^k-1} \hat{\alpha}_t^k(i)\,a_{ij}\,b_j(O_{t+1}^k)\,\hat{\beta}_{t+1}^k(j)\,c_{t+1}^k}{\sum_{k=1}^{K}\sum_{t=1}^{T^k-1} \hat{\alpha}_t^k(i)\,\hat{\beta}_t^k(i)},$$

$$\overline{b}_i(m) = \frac{\sum_{k=1}^{K}\sum_{t:O_t^k=m} \hat{\alpha}_t^k(i)\,\hat{\beta}_t^k(i)}{\sum_{k=1}^{K}\sum_{t=1}^{T^k} \hat{\alpha}_t^k(i)\,\hat{\beta}_t^k(i)}.$$

Erweitertes HMM mit diskreten Ausgaben (Ausgabe-Klassen)

$$\overline{\pi}_i = \frac{\sum_{k=1}^{K} \hat{\alpha}_1^k(i)\,\hat{\beta}_1'^k(i)}{\sum_{k=1}^{K}\sum_{j=1}^{N} \hat{\alpha}_1^k(j)\,\hat{\beta}_1'^k(j)},$$

$$\bar{a}_{ilj} = \frac{\sum_{k=1}^{K} \sum_{\substack{t=1 \\ p_t^k=l}}^{T^k-1} \hat{\alpha}_t^k(i)\, a_{ilj}\, b_j(O_{t+1}^k)\, \hat{\beta}'^k_{t+1}(j)\, c_{t+1}^k}{\sum_{k=1}^{K} \sum_{\substack{t=1 \\ p_t^k=l}}^{T^k-1} \hat{\alpha}_t^k(i)\, \hat{\beta}'^k_t(i)},$$

$$\bar{b}_{jm} = \frac{\sum_{k=1}^{K} \sum_{t:O_t^k=m} \hat{\alpha}_t^k(j)\, \hat{\beta}'^k_t(j)}{\sum_{k=1}^{K} \sum_{t=1}^{T^k} \hat{\alpha}_t^k(j)\, \hat{\beta}'^k_t(j)},$$

mit $p_t^k = cl(\sum_{\tau=1}^{t} O_\tau^k)$.

HMM mit stetigen Ausgaben

$$\bar{\pi}_i = \frac{\sum_{k=1}^{K} \hat{\alpha}_1^k(i)\, \hat{\beta}_1^k(i)}{\sum_{k=1}^{K} \sum_{j=1}^{N} \hat{\alpha}_1^k(j)\, \hat{\beta}_1^k(j)},$$

$$\bar{a}_{ij} = \frac{\sum_{k=1}^{K} \sum_{t=2}^{T^k} \hat{\alpha}_{t-1}^k(i)\, a_{ij} \left[\sum_{m=1}^{M} c_{jm} b_{jm}(O_t^k)\right] \hat{\beta}_t^k(j)\, c_t^k}{\sum_{k=1}^{K} \sum_{t=2}^{T^k} \hat{\alpha}_{t-1}^k(i)\, \hat{\beta}_{t-1}^k(i)},$$

$$\bar{c}_{im} = \frac{\sum_{k=1}^{K} \sum_{t=1}^{T^k} \left[\sum_{j=1}^{N} \hat{\alpha}_{t-1}^k(j)\, a_{ji}\right] c_{im}\, b_{im}(O_t^k)\, \hat{\beta}_t^k(i)\, c_t^k}{\sum_{k=1}^{K} \sum_{t=1}^{T^k} \hat{\alpha}_t^k(i)\, \hat{\beta}_t^k(i)},$$

$$\bar{\mu}_{im} = \frac{\sum_{k=1}^{K} \sum_{t=1}^{T^k} \left[\sum_{j=1}^{N} \hat{\alpha}_{t-1}^k(j)\, a_{ji}\right] c_{im}\, b_{im}(O_t^k)\, \hat{\beta}_t^k(i)\, c_t^k\, O_t^k}{\sum_{k=1}^{K} \sum_{t=1}^{T^k} \left[\sum_{j=1}^{N} \hat{\alpha}_{t-1}^k(j)\, a_{ji}\right] c_{im}\, b_{im}(O_t^k)\, \hat{\beta}_t^k(i)\, c_t^k},$$

$$\bar{u}_{im} = \frac{\sum_{k=1}^{K} \sum_{t=1}^{T^k} \left[\sum_{j=1}^{N} \hat{\alpha}_{t-1}^k(j)\, a_{ji}\right] c_{im}\, b_{im}(O_t^k)\, \hat{\beta}_t^k(i)\, c_t^k\, (O_t^k - \mu_{im})^2}{\sum_{k=1}^{K} \sum_{t=1}^{T^k} \left[\sum_{j=1}^{N} \hat{\alpha}_{t-1}^k(j)\, a_{ji}\right] c_{im}\, b_{im}(O_t^k)\, \hat{\beta}_t^k(i)\, c_t^k}.$$

Für $t = 1$ ist der Ausdruck $\left[\sum_{j=1}^{N} \hat{\alpha}_{t-1}^{k}(j)\, a_{ji}\right]$ in den Gleichungen der $\bar{c}_{im}$, $\bar{\mu}_{im}$ und $\bar{u}_{im}$ durch π_i zu ersetzen ist, da die $\hat{\alpha}_0^k(j)$ nicht definiert sind.

Erweitertes HMM mit stetigen Ausgaben (Ausgabe-Klassen)

$$\bar{\pi}_i = \frac{\sum_{k=1}^{K} \hat{\alpha}_1^k(i)\, \hat{\beta}_1'^k(i)}{\sum_{k=1}^{K}\sum_{j=1}^{N} \hat{\alpha}_1^k(j)\, \hat{\beta}_1'^k(j)},$$

$$\bar{a}_{ij} = \frac{\sum_{k=1}^{K}\sum_{\substack{t=2 \\ p_t^k=l}}^{T^k} \hat{\alpha}_{t-1}^k(i)\, a_{ilj} \left[\sum_{m=1}^{M} c_{jm} b_{jm}(O_t^k)\right] \hat{\beta}_t'^k(j)\, c_t^k}{\sum_{k=1}^{K}\sum_{\substack{t=2 \\ p_t^k=l}}^{T^k} \hat{\alpha}_{t-1}^k(i)\, \hat{\beta}_{t-1}'^k(i)},$$

$$\bar{c}_{im} = \frac{\sum_{k=1}^{K}\sum_{t=1}^{T^k} \left[\sum_{j=1}^{N} \hat{\alpha}_{t-1}^k(j)\, a_{jp_{t-1}i}\right] c_{im}\, b_{im}(O_t^k)\, \hat{\beta}_t'^k(i)\, c_t^k}{\sum_{k=1}^{K}\sum_{t=1}^{T^k} \hat{\alpha}_t^k(i)\, \hat{\beta}_t'^k(i)},$$

$$\bar{\mu}_{im} = \frac{\sum_{k=1}^{K}\sum_{t=1}^{T^k} \left[\sum_{j=1}^{N} \hat{\alpha}_{t-1}^k(j)\, a_{jp_{t-1}i}\right] c_{im}\, b_{im}(O_t^k)\, \hat{\beta}_t'^k(i)\, c_t^k\, O_t^k}{\sum_{k=1}^{K}\sum_{t=1}^{T^k} \left[\sum_{j=1}^{N} \hat{\alpha}_{t-1}^k(j)\, a_{jp_{t-1}i}\right] c_{im}\, b_{im}(O_t^k)\, \hat{\beta}_t'^k(i)\, c_t^k},$$

$$\bar{u}_{im} = \frac{\sum_{k=1}^{K}\sum_{t=1}^{T^k} \left[\sum_{j=1}^{N} \hat{\alpha}_{t-1}^k(j)\, a_{jp_{t-1}i}\right] c_{im}\, b_{im}(O_t^k)\, \hat{\beta}_t'^k(i)\, c_t^k\, (O_t^k - \mu_{im})^2}{\sum_{k=1}^{K}\sum_{t=1}^{T^k} \left[\sum_{j=1}^{N} \hat{\alpha}_{t-1}^k(j)\, a_{jp_{t-1}i}\right] c_{im}\, b_{im}(O_t^k)\, \hat{\beta}_t'^k(i)\, c_t^k},$$

mit

$$p_t^k = cl\left(\sum_{\tau=1}^{t} O_\tau^k\right).$$

Für $t = 1$ ist der Ausdruck $\left[\sum_{j=1}^{N} \hat{\alpha}_{t-1}^k(j)\, a_{jp_{t-1}i}\right]$ in den Gleichungen der $\bar{c}_{im}$, $\bar{\mu}_{im}$ und $\bar{u}_{im}$ durch π_i zu ersetzen, da die $\hat{\alpha}_0^k(j)$ und $a_{jp_0 i}$ nicht definiert sind.

Anhang B

Grenzwerte der Funktion $p(\mu)$

Es sei $p(\mu)$ die in Abschnitt 5.2.2 eingeführte Funktion

$$p(\mu) := E - \mu - (V - \mu E)\, \bar{f}(0; \mu, \sqrt{V - \mu E}) = 0$$

mit den Konstanten

$$\begin{aligned} E &= \frac{\sum_{t=1}^{T} \zeta_t O_t}{\sum_t \zeta_t}\,, \\ V &= \frac{\sum_{t=1}^{T} \zeta_t O_t^2}{\sum_t \zeta_t}\,, \\ &\quad \text{mit } \; 0 \le \zeta_t, O_t \le K \;\text{ und }\; E, V > 0\,, \end{aligned}$$

und der Dichte der linksseitig bei $x = 0$ gestutzten Normalverteilung (vgl. Abschnitt 5.2.1)

$$\bar{f}(x; \mu, \sigma) = \frac{1}{\sigma \Phi(\mu/\sigma)}\, \varphi(z) \qquad \text{mit } \; z = \frac{x - \mu}{\sigma}\,,\; x \ge 0\,,$$

wobei $\varphi(\cdot)$ und $\Phi(\cdot)$ die Dichte und die Verteilungsfunktion der standardisierten Normalverteilung bezeichnen. Für die Parameter gelte

$$-\infty < \mu < \infty\,,\; 0 < \sigma < \infty\,.$$

Satz B.1 *Für $\mu \to -\infty$ strebt $p(\mu)$ gegen 0 und für $\mu \to V/E$ gilt $p(\mu) \to E - V/E$.*

Beweis: Für $x > 0$ gilt die konvergente Kettenbruchentwicklung [35]

$$\Phi(x) = 1 - \cfrac{\varphi(x)}{x + \cfrac{1}{x + \cfrac{2}{x + \cfrac{3}{\cdots}}}}\,. \tag{B.1}$$

Wir setzen $\sigma^2 := V - \mu E$ und $\nu := \mu/\sigma$, dann gilt für $\mu \neq 0$: $\sigma = \mu/\nu$ und

$$p(\mu) = E - \mu - \sigma \frac{\varphi(\mu/\sigma)}{\Phi(\mu/\sigma)} = E - \mu \left[1 + \frac{1}{\nu} \frac{\varphi(\nu)}{\Phi(\nu)}\right]. \tag{B.2}$$

1.) Sei $\mu < 0$, dann folgt mit (B.1)

$$\Phi(\nu) = 1 - \Phi(-\nu) = \cfrac{\varphi(\nu)}{-\nu + \cfrac{1}{-\nu + \cfrac{2}{\cdots}}},$$

und damit

$$\frac{1}{\nu} \frac{\varphi(\nu)}{\Phi(\nu)} = -1 + \cfrac{1}{-\nu^2 + \cfrac{2}{-1 + \cfrac{3}{-\nu^2 + \cdots}}}.$$

Durch Rücktransformation von ν und σ erhalten wir schließlich

$$\begin{aligned}
p(\mu) &= E - \cfrac{\mu}{\cfrac{-\mu^2}{V - \mu E} + \cfrac{2}{-1 + \cfrac{3}{\cfrac{-\mu^2}{V - \mu E} + \cfrac{4}{\cdots}}}} \\
&= E - \cfrac{1}{\cfrac{-\mu}{V - \mu E} + \cfrac{2}{-\mu + \cfrac{3}{\cfrac{-\mu}{V - \mu E} + \cfrac{4}{\cdots}}}} \\
&= E - \cfrac{1}{\cfrac{-1}{\frac{V}{\mu} - E} + \cfrac{2}{-\mu + \cfrac{3}{\cfrac{-1}{\frac{V}{\mu} - E} + \cfrac{4}{\cdots}}}} \\
&\xrightarrow{\mu \to -\infty} E - E = 0.
\end{aligned}$$

2.) Für $\mu \to V/E$ geht $\sigma^2 = V - \mu E$ gegen 0, und mit (B.2) gilt

$$\begin{aligned}
\lim_{\mu \to \frac{V}{E}} p(\mu) &= \lim_{\sigma \to 0} E - \frac{V}{E} - \sigma \frac{\varphi(\frac{V}{\sigma E})}{\Phi(\frac{V}{\sigma E})} \\
&= E - \frac{V}{E}.
\end{aligned}$$

□

Literaturverzeichnis

[1] P. Baldi und Y. Chauvin. Smooth on-line learning algorithms for hidden Markov models. *Neural Computation*, 6:307–318, 1994.

[2] L. E. Baum. An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes. In *Inequalities, III*, Seiten 1–8. Academic Press, New York, 1972.

[3] L. E. Baum, T. Petrie, G. Soules und N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Ann. Math. Statist.*, 41:164–171, 1970.

[4] C. Becchetti und L. Prina Ricott. *Speech Recognition: Theory und C++ Implementation*. John Wiley & Sons, New York, 1999.

[5] E. Bertsch, B. Hölzle und H. Laux, Herausgeber. *Handwörterbuch der Bauspartechnik.* Verlag Versicherungswirtschaft GmbH, Karlsruhe, 1998.

[6] J. A. Bilmes. A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. Technischer Bericht TR-97-021, International Computer Science Institute, Berkeley, CA, 1998.

[7] A. P. Dempster, N. M. Laird und D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Statist. Soc. Ser. B*, 39(1):1–38, 1977.

[8] A. P. Dunmur und D. M. Titterington. The influence of initial conditions on maximum likelihood estimation of the parameters of a binary hidden Markov model. *Statist. Probab. Lett.*, 40(1):67–73, 1998.

[9] R. Durbin, S. Eddy, A. Krogh und G. Mitchison. *Biological sequence analysis. Probabilistic models of proteins and nucleic acids.* Cambridge University Press, 1998.

[10] S. R. Eddy. Hidden Markov models. *Current Opinion in Structural Biology*, 6:361–365, 1996.

[11] B.S. Everitt. *Cluster Analysis.* Edward Arnold, London, 1993.

[12] G. S. Fishman. *Monte Carlo – Concepts, Algorithms and Applications.* Springer, New York, 1996.

[13] J. L. Gauvain und C.-H. Lee. Bayesian learning for hidden Markov models with Gaussian mixture state observation densities. *Speech communication*, 11(2-3):205–214, 1992.

[14] Z. Ghahramani und M. I. Jordan. Factorial hidden Markov models. *Machine Learning*, 29:245–273, 1997.

[15] F. Gotterbarm. *Modelle und Optimierungsansätze zur Analyse des kollektiven Bausparens*. Dissertation, Universität Bonn, Bonn, 1985.

[16] J. Hartung. *Statistik: Lehr- und Handbuch der angewandten Statistik*. R. Oldenbourg Verlag, München, 1982.

[17] X. D. Huang, Y. Ariki und M. A. Jack. *Hidden Markov Models for Speech Recognition*. Edinburgh University Press, 1990.

[18] A. K. Jain und R. C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, Englewood Cliffs, New Jersey, 1988.

[19] N. L. Johnson und S. Kotz. *Distributions in statistics: continuous univariate distributions*, Band 1,2. Houghton Mifflin Company, Boston, 1970.

[20] B.-H. Juang. Maximum-likelihood estimation for mixture multivariate stochastic observations of Markov chains. *AT&T Tech. J.*, 64(6, Teil 1):1235–1249, 1985.

[21] B.-H. Juang, S. E. Levinson und M. M. Sondhi. Maximum likelihood estimation for multivariate mixture observations of Markov chains. *IEEE Trans. Informat. Theory*, IT-32(2):307–309, 1986.

[22] B.-H. Juang und L. R. Rabiner. A probabilistic distance measure for hidden Markov models. *AT&T Tech. J.*, 64(2, Teil 1):391–408, 1985.

[23] M. H. Kalos und P. A. Whitlock. *Monte Carlo Methods*. John Whiley & Sons, New York, 1986.

[24] I. Kellershohn. *Mathematische Simulation von Bausparkollektiven mit Hilfe von empirischen Verteilungen in monothetischen hierarchischen Kollektivclusterungen*. Dissertation, Universität zu Köln, Köln, 1992.

[25] B. Knab, R. Schrader, I. Weber, K. Weinbrecht und B. Wichern. Mesoskopisches Simulationsmodell zur Kollektivfortschreibung. Technischer Bericht ZPR97-295, Mathematisches Institut, Universität zu Köln, 1997.

[26] Bundesgeschäftsstelle Landesbausparkassen, Herausgeber. *Bausparkassen-Fachbuch 1997*. Deutscher Sparkassenverlag GmbH, Stuttgart, 1997.

[27] P. Langrock und W. Jahn. *Einführung in die Teorie der Markovschen Ketten und ihre Anwendungen*. BSB B.G. Teubner Verlagsgesellschaft, Leipzig, 1979.

[28] S. E. Levinson, L. R. Rabiner und M. M. Sondhi. An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition. *Bell System Tech. J.*, 62(4):1035–1074, 1983.

[29] L. A. Liporace. Maximum likelihood estimation for multivariate observations of Markov sources. *IEEE Trans. Inform. Theory*, 28(5):729–734, 1982.

[30] R. B. Lyngsø, C. N. S. Pedersen und H. Nielsen. Measures an hidden Markov models. Technischer Bericht RS-99-6, BRICS, 1999.

[31] I. L. MacDonald und W. Zucchini. *Hidden Markov and Other Models for Descrete-valued Time Series.* Chapman & Hall, London, 1997.

[32] G. J. McLachlan und K. E. Basford. *Mixture Models: Inference and Applications to Clustering.* Marcel Dekker, Inc., New York, Basel, 1988.

[33] G. J. McLachlan und T. Krishnan. *The EM Algorithm and Extensions.* John Wiley & Sons, Inc, New York, 1997.

[34] Ralph Mermagen. Evaluierung von Clusteralgorithmen für Bausparkollektive. Diplomarbeit, Köln, 1995.

[35] P. H. Müller, Herausgeber. *Wahrscheinlichkeitsrechnung und Mathematische Statistik – Lexikon der Stochastik.* Akademie Verlag, Berlin, 1991.

[36] J. R. Norris. *Markov Chains.* Cambridge University Press, 1997.

[37] W. H. Press, S. A. Teukolsky, W. T. Vetterling und B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge University Press, 1993.

[38] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–285, 1989.

[39] L. R. Rabiner, B.-H. Juang, S. E. Levinson und M. M. Sondhi. Some properties of continuous hidden Markov model representations. *AT&T Tech. J.*, 64(6, Teil 1):1251–1270, 1985.

[40] L. R. Rabiner, S. E. Levinson und M. M. Sondhi. On the application of vector quantization and hidden Markov models to speaker-independent, isolated word recognition. *Bell System Tech. J.*, 62(4):1075–1105, 1983.

[41] A. Stolcke und S. M. Omohundro. Best-first model merging for hidden Markov model induction. Technischer Bericht TR-94-003, International Computer Science Institute, Berkeley, CA, January 1994.

[42] S. Thrun und J. Langford. Monte carlo hidden Markov models. Technischer Bericht CMU-CS-98-179, Carnegie Mellon University, Pittsburgh, PA, 1998.

[43] H. Trebbe. The Münster tagging project – errata in Rabiner's HMM-tutorial. Arbeitsbereich Linguistik, Westfälische Wilhelms-Universität, Münster, 1995.

[44] I. Vannahme. *Clusteralgorithmen zur mathematischen Simulation von Bausparkollektiven.* Dissertation, Universität zu Köln, Köln, 1996.

Danksagung

Diese Arbeit wurde durch die Unterstützung von vielen Seiten vorangetrieben. Besonders bedanken möchte ich mich bei

- Prof. Dr. R. Schrader und Prof. Dr. A. Bachem für die Gelegenheit, am ZPR/ZAIK in einem motivierten Umfeld praxisorientiert und selbsständig zu arbeiten und zu forschen;
- Alexander Schliep, der immer als Ansprechpartner für mich da war und dieser Arbeit entscheidende Impulse gegeben hat;
- Bernd Wichern für viele Diskussionen, Anregungen, Kritiken und Korrekturen;
- Dr. Barthel Steckemetz für die geistige Geburtshilfe;
- Dr. Georg Bauer, Dr. Jörg Schepers und Dr. Karin Weinbrecht für kritisches Korrekturlesen;
- Iris Weber, Thomas Chevalier, Alexander Schönhuth und Dirk Räbiger aus der Bausparkassengruppe für die gute Zusammenarbeit und die angenehme Arbeitsatmosphäre;
- allen weiteren Kolleginnen und Kollegen für ihre ständige Hilfsbereitschaft und die schöne Zeit am ZPR/ZAIK;
- meinen Eltern, die mir in der Wahl des Berufswegs völlige Freiheit ließen und immer hinter mir standen;
- meiner Freundin Tanja Soehnlen für ihre immer währende moralische Unterstützung und Geduld.

Lebenslauf

Persönliche Daten

Name:	Bernhard Knab
Adresse:	Rheinbacher Straße 14, 50937 Köln
Geburtsdatum:	15. September 1967
Geburtsort:	Worms
Familienstand:	ledig
Staatsangehörigkeit:	deutsch

Ausbildung/Wehrdienst/Studium

1973–1977	Paternus-Grundschule, Worms-Pfeddersheim
1977–1986	Rudi-Stephan-Gymnasium Worms, Abschluss Abitur
1986–1987	Grundwehrdienst, Westerburg
1987–1994	Studium der Technomathematik, Universität Karlsruhe
9/1989	Vordiplom in Technomathematik
1990–1991	Mathematik-Studium an der Université Joseph Fourier, Grenoble (Frankreich); Abschluss: Licence de mathématiques
9/1994	Diplom in Technomathematik

Berufstätigkeit

11/1994–3/1995	Wissenschaftliche Hilfskraft am Institut für Mechanische Verfahrenstechnik und Mechanik, Universität Karlsruhe
4/1995–3/2000	Wissenschaftlicher Mitarbeiter am Mathematischen Institut / Zentrum für Angewandte Informatik Köln, Universität zu Köln

Köln, den 23. Mai 2000

Erklärung

Ich versichere, dass ich die von mir vorgelegte Dissertation selbständig angefertigt, die benutzten Quellen und Hilfsmittel vollständig angegeben und die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken im Wortlaut oder dem Sinn nach entnommen sind, in jedem Einzelfall als Entlehnung kenntlich gemacht habe; dass diese Dissertation noch keiner anderen Fakultät oder Universität zur Prüfung vorgelegen hat; dass sie noch nicht veröffentlicht worden ist sowie, dass ich eine solche Veröffentlichung vor Abschluss des Promotionsverfahrens nicht vornehmen werde. Die Bestimmungen der Promotionsordnung sind mir bekannt. Die von mir vorgelegte Promotion ist von Professor Dr. R. Schrader betreut worden.