Hidden-Markov-Modelle zur Analyse und Simulation von Finanzzeitreihen

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

vorgelegt von
Bernd Wichern
aus Sittensen

Hundt Druck GmbH, Köln

Köln 2001

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

Tag der mündlichen Prüfung: 8. November 2001

Inhaltsverzeichnis

Kapitel 1

Einleitung

Hidden-Markov-Modelle (HMM) werden seit etwa zwei Jahrzehnten sehr erfolgreich in ganz unterschiedlichen Anwendungsgebieten der stochastischen Modellierung eingesetzt. Zu den wichtigsten zählen die automatische Spracherkennung, das Lösen verschiedener Probleme im Bereich der Bioinformatik, die Mustererkennung und die Analyse von Zeitreihen. Ein Grund für die vielfältigen Einsatzmöglichkeiten von Hidden-Markov-Modellen ist darin zu sehen, dass es sich zwar um einen sehr allgemeinen Modellansatz handelt, der aber dennoch durch die Vorgabe spezifischer Eigenschaften sehr gut an das zu lösende Problem angepasst werden kann.

Die Gemeinsamkeit aller HMM-Anwendungen liegt darin, dass reale Daten eingesetzt werden können, um die Modellparameter in einem automatisierten Verfahren anzupassen (Training des Modells). Mit dem Baum-Welch-Algorithmus existiert ein effizienter Trainingsalgorithmus, der mit zur großen Verbreitung der Hidden-Markov-Modelle beigetragen hat.

In der vorliegenden Arbeit wird eine neue Anwendung für Hidden-Markov-Modelle entwickelt. Der Gegenstand ist die Modellierung und Simulation relevanter Zeitreihen eines Bausparkollektives. Der Einsatz verlässlicher Simulationsmodelle ist für eine Bausparkasse ein wichtiges Werkzeug zur Analyse der zukünftigen Ertragslage und zum Risikomanagement. Es ist zu erwarten, dass die Bedeutung von Simulationen im Bausparwesen in naher Zukunft aufgrund der Einführung verschärfter Kontrollmechanismen noch wachsen wird.

Der Kern des hier entwickelten Hidden-Markov-Bauspar-Modells (HMBM) ist ein modellbasierter Clusteralgorithmus, mit dessen Hilfe die stochastischen Eigenschaften eines Bausparkollektivs parametrisiert werden. Das Verfahren basiert auf den fein aufgelösten Daten einzelner Bausparverträge, die je nach Alter auf unterschiedlich lange Zeitreihen abgebildet werden. Die HMM-Modellierung eignet sich ideal zur Abbildung dieser unterschiedlich dimensionierten Objekte, was einen großen Vorteil gegenüber herkömmlichen geometrischen Clusterverfahren darstellt.

Jedes Bausparsimulationsmodell muss die beiden folgenden Aufgaben lösen:

- Integration bestehender Verträge (Bestandsabbildung)
- Generierung zukünftiger Zeitreihen

Der Reiz des in dieser Arbeit entwickelten Modells liegt darin, dass diese beiden Aspekte – ähnlich wie in einem realen Bausparkollektiv – sehr eng miteinander verknüpft werden. Die trainierten Hidden-Markov-Modelle bilden den Bestand ab und können direkt unter Verwendung eines Zufallszahlengenerators zur Fortschreibung bestehender und zur Generierung künstlicher Zeitreihen eingesetzt werden. Aus der Überlagerung dieser Zeitreihen können schließlich Aussagen über die zukünftige Entwicklung wichtiger Kollektivgrößen gemacht werden. In [23] wurde die Basis des hier verwendeten Modellansatzes entwickelt und verschiedene der dort vorgestellten Ideen sind auch Grundlage dieser Arbeit. Im Folgenden werden kurz die Inhalte der einzelnen Kapitel erläutert.

Im Kapitel zwei werden nach einer Literaturübersicht eine Einführung in die Grundlagen der Theorie kontinuierlicher Hidden-Markov-Modelle und eine Darlegung der wichtigsten Algorithmen im Bereich der Hidden-Markov-Modelle gegeben. Zusätzlich werden verschiedene Erweiterungen, die für einen erfolgreichen Einsatz in der Bausparmodellierung notwendig sind, erläutert.

Das Kapitel drei stellt einen modellbasierten Clusteralgorithmus zur Clusterung komplexer Daten vor. Es handelt sich um die Verknüpfung von Hidden-Markov-Modellen mit dem Ansatz der Mischmodellierung, deren Stärke in der Clusterung von Daten ohne klare Klasseneinteilung liegt.

Im Kapitel vier wird nach einer Erläuterung der elementaren Bausparbegriffe und des generellen Ablaufs eines Bausparvertrages das vollständige HMBM vorgestellt. Es beschreibt den Modellierungsansatz und die Erzeugung der Zeitreihen aus dem zur Verfügung stehenden Datensatz. Außerdem werden die relevanten Charakteristika des Hidden-Markov-Modells zum Erreichen einer guten Abbildung der Bauspardaten erläutert. Dies beinhaltet verschiedene Erweiterungen, die gegenüber dem Standard-HMM notwendig sind.

Das Training ist der zentrale Punkt in allen HMM-Anwendungen und eine gute Parameterbestimmung ist die notwendige Voraussetzung für die Durchführung einer Bausparsimulation. Daher werden im Kapitel fünf verschiedene Aspekte des Modelltrainings anhand realer Daten untersucht. Es werden Kriterien zur Bewertung von Cluster- und Trainingsergebnissen entwickelt, mit deren Hilfe unterschiedliche Varianten des Modelltrainings beurteilt werden. Als zwei wesentliche Punkte, die betrachtet werden, seien die Zusammensetzung der Trainingsmenge und die Modelltopologie erwähnt.

Im Kapitel sechs werden verschiedene Untersuchungen, die die Simulation betreffen, dargestellt. Das beinhaltet die Generierung künstlicher Sequenzen und die Verlängerung realer Anfangssequenzen. In diesem Zusammenhang wird auch das Zuordnungsproblem untersucht: Um eine Anfangsteilsequenz unter Verwendung eines Generatormodells zu vervollständigen, muss entschieden werden, welchem der zur Auswahl stehenden Modelle die Sequenz am sinnvollsten zuzuordnen ist. Im Anschluss hieran wird beschrieben, wie aus den generierten Zeitreihen ein komplettes Bausparkollektiv erzeugt wird. Abschließend wird eine Bewertung des HMBM und der Simulationsergebnisse vorgenommen.

Die Arbeit endet mit einer Zusammenfassung der Ergebnisse und einem Ausblick auf mögliche Weiterentwicklungen des Modells.

Kapitel 2

Hidden-Markov-Modelle (HMM)

Diese Kapitel gibt einen umfassenden Überblick über das Konzept der Hidden-Markov-Modelle, ein Modellansatz, der zur stochastischen Modellierung von Daten in verschiedenen Anwendungsgebieten sehr erfolgreich eingesetzt wird. Im ersten Abschnitt wird eine Übersicht zentraler Arbeiten über Hidden-Markov-Modelle sowohl im Feld der Anwendung als auch in der Theorie gegeben. Das darauf folgende Kapitel beschreibt die grundlegenden Elemente von Hidden-Markov-Modellen und liefert damit gleichzeitig eine Zusammenfassung der in dieser Arbeit verwendeten Notation. Im dritten Unterkapitel werden verschiedene HMM-Algorithmen präsentiert. Hierbei handelt sich um den Forward-Backward-Algorithmus, den Viterbi-Algorithmus, den Baum-Welch-Algorithmus und um Algorithmen zur Klassifikation und zum Clustern von Sequenzen. Im Anschluss daran werden einige Erweiterungen von Hidden-Markov-Modellen beschrieben, die in der Modellierung der Bauspardaten eine wichtige Rolle spielen. Das Kapitel schließt mit einem Abschnitt über den EM-Algorithmus. Dieses grundlegende und allgemeine Verfahren liefert die theoretische Rechtfertigung des Baum-Welch-Algorithmus und spielt eine entscheidende Rolle in der Modellierung mit so genannten Mischmodellen, die im Kapitel 3 ausführlich behandelt werden.

2.1 Theorie und Anwendungen: Ein Überblick

Hidden-Markov-Modelle haben in den letzten zwei Jahrzehnten auf der einen Seite eine stetig wachsende Bedeutung in verschiedensten Anwendungsgebieten gewonnen; auf der anderen Seite waren sie auch Gegenstand intensiver theoretischer Untersuchungen. Bevor im nächsten Abschnitt Hidden-Markov-Modelle durch eine Zusammenstellung wichtiger Begriffe und Notationen genauer beschrieben werden, soll hier zunächst ein kurzer Überblick über die HMM-Literatur gegeben werden.

Hidden-Markov-Modelle wurden erstmals Ende der sechziger, Anfang der siebziger Jahre in [3] und [2] – dort und allerdings noch unter der Bezeichnung „probabilistische Funktionen von Markov-Ketten“ – analysiert. In den genannten Arbeiten wird bereits das iterative Verfahren zur Maximum-Likelihood-Schätzung, später mit dem Namen Baum-Welch-Algorithmus versehen,

entwickelt. Anfang der achtziger Jahre wurden verschiedene Erweiterungen des Verfahrens auf allgemeinere Klassen von Ausgabefunktionen entwickelt [28], [18]. Eine neuere, umfassende theoretische Einführung in das Gebiet der Hidden-Markov-Modelle mit Anwendungen in der Kontrolltheorie liegt mit [9] vor.

Ende der siebziger und Anfang der achtziger Jahre erschienen auch die ersten Arbeiten zu Anwendungen der Hidden-Markov-Modelle in der automatischen Spracherkennung, wie z. B. [27]. Dieser Anwendungsbereich hat seitdem eine sehr starke Weiterentwicklung erfahren und ist mittlerweile industrieller Standard, was ohne den Einsatz von Hidden-Markov-Modellen wohl kaum möglich gewesen wäre. Mit [32] existiert ein sehr gelungenes Tutorium über Hidden-Markov-Modelle , das ebenfalls auf der automatischen Spracherkennung basiert.

In der Folge kamen diverse Anwendungsgebiete hinzu. Besonders in dem Gebiet der Bioinformatik werden Hidden-Markov-Modelle zur Lösung verschiedener Probleme eingesetzt [25]. Auch in der Mustererkennung in Zeitreihen [33] und in der Modellierung von Finanzzeitreihen [11] findet dieser Modellierungsansatz seine Anwendung.

2.2 Modellbeschreibung und Notationen

Markov-Kette und Hidden-Markov-Modell
Ein Hidden-Markov-Modell ist ein stochastisches Modell und kann als eine Erweiterung einer einfachen Markov-Kette betrachtet werden. Eine Markov-Kette beschreibt ein System, das sich in diskreten Zeitschritten fortentwickelt. Das System kann eine endliche Anzahl von verschiedenen Zuständen einnehmen und zu jedem Zeitpunkt befindet es sich in genau einem davon. Der Wechsel zwischen den Zuständen erfolgt stochastisch und wird durch Übergangswahrscheinlichkeiten bestimmt. Bei einer Markov-Kette erster Ordnung ist die Wahrscheinlichkeit des Vorliegens eines bestimmten Zustandes zum Zeitpunkt t lediglich abhängig davon, in welchem Zustand sich das System zum Zeitpunkt $t-1$ befunden hat. Die vor $t-1$ eingenommenen Zustände sind quasi „vergessen“. Dies wird auch als Markov-Eigenschaft bezeichnet.

Eine Markov-Kette n-ter Ordnung ist eine Verallgemeinerung hiervon, dahingehend dass bei ihr n Vorgängerzustände relevant sind und in die Berechnung von Übergangswahrscheinlichkeiten eingehen. In dieser Arbeit werden jedoch ausschließlich Markov-Ketten erster Ordnung betrachtet.

Bei einer Markov-Kette handelt es sich um einen so genannten stochastischen Prozess. Die Erweiterung zum Hidden-Markov-Modell besteht darin, dass jeder Zustand mit einer Verteilungsfunktion verknüpft wird. Im Fall einer diskreten Verteilungsfunktion nennt man das entsprechende Modell diskretes Hidden-Markov-Modell. Im kontinuierlichen Fall spricht man von einem kontinuierlichen Hidden-Markov-Modell.

In jedem Zeitschritt gibt das System ein Symbol gemäß der Wahrscheinlichkeitsverteilung des Zustandes aus, in dem es sich gerade befindet und wechselt anschließend gemäß der Übergangswahrscheinlichkeiten in den nächsten Zustand. Die zeitliche Abfolge der ausgegebenen Symbole wird generell als Sequenz bezeichnet. Lediglich die ausgegebenen Symbole sind einem Beobachter zugänglich, die darunter liegenden Zustände, die zur Ausgabe dieser Sequenz geführt

haben, bleiben verborgen, womit sich auch das Wort „Hidden“ im Namen der hier untersuchten Modelle erklärt. Ein Hidden-Markov-Modell kann als eine Kopplung zweier Zufallsprozesse angesehen werden:

1. Entsprechend der Übergangswahrscheinlichkeiten wird ein Zustand zufällig gewählt.
2. Ein Ausgabesymbol wird zufällig gemäß der Wahrscheinlichkeitsverteilung des Zustandes bestimmt.

Ein Beispiel
Bevor weiter unten die formalen Definitionen zur Beschreibung von Hidden-Markov-Modellen präsentiert werden, soll zunächst ein sehr einfaches, intuitives Beispiel zur Veranschaulichung eines diskreten HMM gegeben werden. Dieses Beispiel geht auf Rabiner [32] zurück.

Es wird folgendes Zufallsexperiment durchgeführt: Gegeben sind N Urnen, in jeder befindet sich eine unbekannte, große Zahl farbiger Bälle. Insgesamt gibt es M mögliche Farben, wobei die gleiche Farbe in einer Urne mehrfach vorkommen kann und keine der Urnen leer ist. Zu jedem Zeitpunkt wird aus einer der Urnen ein Ball gezogen, die Farbe wird registriert und der Ball wird in die Urne zurückgelegt. Die Auswahl der Urnen geschieht durch einen iterativen Zufallsprozess so, dass die Wahl der aktuellen Urne nur von der Vorgängerurne abhängt. Die erste Urne wird ebenfalls zufällig ausgewählt. Auf die Art und Weise entsteht eine Sequenz von Farben, die korrespondierende Sequenz von Urnen wird jedoch nicht festgehalten.

Wie sieht ein möglichst einfaches Modell aus, das dieses Zufallsexperiment erklären könnte? Für jede der Urnen wird ein Zustand eingeführt. Mit dem Zustand wird eine diskrete Wahrscheinlichkeitsverteilung entsprechend der Häufigkeiten der einzelnen Farben in der Urne assoziiert. Zusätzlich wird eine Übergangsmatrix für die Zustände eingesetzt, die die Auswahl der Urnen beschreibt. Außerdem wird eine weitere Verteilung verwendet, die die Auswahl der ersten Urne im probabilistischen Sinne festlegt. Dieses Modell ist sehr gut geeignet, das angeführte Zufallsexperiment zu simulieren und trotz der Einfachheit enthält es bereits alle Elemente eines diskreten Hidden-Markov-Modells. Bei den Elementen, die ein HMM definieren, kann zwischen der Modelltopologie und den Modellparametern unterschieden werden, wenngleich beide Bereiche voneinander abhängen.

Modelltopologie
Die Topologie (synonym wird häufig auch der Begriff Struktur verwendet) eines Hidden-Markov-Modells, die von außen vorgegeben und in der Regel nicht verändert wird, umfasst die folgenden Elemente:

Zustände: Die Anzahl der Zustände wird vorgegeben und mit N bezeichnet. Die Zustände werden durchnummeriert und mit den Zahlen $(1, \ldots, N)$ gekennzeichnet.

Ausgaben: Für die Ausgabe von Symbolen muss für jeden Zustand eine Dichtefunktion angegeben werden. Die Festlegung der Funktionsfamilie, aus der diese Dichtefunktion stammt, wird der Topologie zugerechnet. Es ist möglich, eine Konvexkombination mehrerer Dichten zu verwenden.

Übergänge: Die Festlegung, welche Zustandsübergänge prinzipiell erlaubt und welche verboten sind, ist ein wichtiges Element der Struktur eines Hidden-Markov-Modells.

Anzahl Übergangsklassen: In den Abschnitten 2.4 und 4.5 werden so genannte Übergangsklassen eingeführt, deren vorzugebende Anzahl ebenfalls ein Bestandteil der Modelltopologie ist.

Die Vorgabe einer Topologie bietet die Möglichkeit, physikalische Eigenschaften des zu modellierenden Prozesses einfließen zu lassen. In vielen Anwendungen werden besonders die möglichen Zustandsübergänge eingeschränkt. So hat es sich in der automatischen Spracherkennung als sinnvoll erwiesen, die Wahl der Modelle auf so genannte „Links-Rechts-Modelle“ einzuschränken [15,27,32]. Ein Beispiel eines solchen Modells zeigt die Abbildung 2.1 in Form eines Graphen.

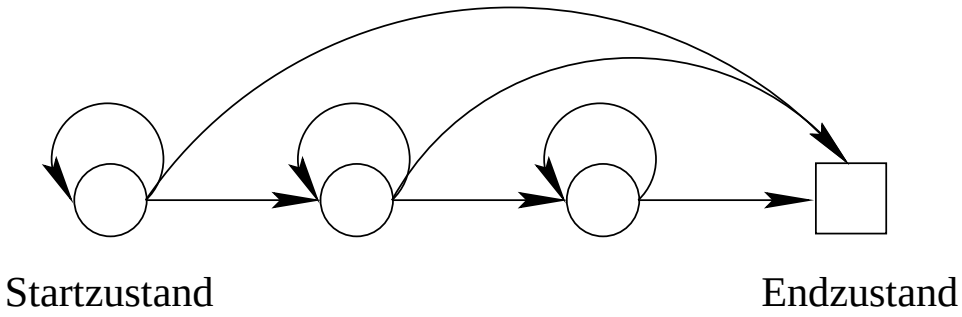

Abbildung 2.1: *Topologie eines einfachen Links-Rechts-Modells*

Jeder der Knoten des Graphen repräsentiert einen Zustand des Modells und die gerichteten Kanten zwischen den Knoten geben die erlaubten Zustandsübergänge wieder. Durch die Wahl dieser Topologie wird dem Modell eine Zeitachse aufgeprägt: Die Zustände sind zeitlich geordnet, jedoch ist durch das Vorhandensein von Selbstübergängen kein eindeutiger Zusammenhang zwischen einem Zustand und einem ausgegebenen Symbol vorhanden. Auch in dieser Arbeit werden zur Modellierung der Bausparzeitreihen Einschränkungen der möglichen Zustandsübergänge vorgenommen, die im Abschnitt 4.4 erklärt werden.

Modellparameter

Unter Modellparameter versteht man im Allgemeinen die Modellgrößen, die vom Training an die Daten angepasst werden. Jedes HMM ist mit einer Menge von Parametern verknüpft, in deren Existenz die große Flexibilität der Hidden-Markov-Modelle begründet liegt. Die Modellparameter sollen durch den Trainingsalgorithmus so bestimmt werden, dass das resultierende Modell die Daten in ihren statistischen Eigenschaften möglichst gut repräsentiert. Details hierüber finden sich im folgenden Abschnitt 2.3 und im Kapitel 5. Unter der Annahme, dass das Modell aus N Zuständen besteht, enthält jedes HMM die folgenden drei Parametersätze:

Initialwahrscheinlichkeiten: Die Wahrscheinlichkeiten, dass sich das System zum Startzeitpunkt in den jeweiligen Zuständen befindet, werden mit $\pi = (\pi_1, \ldots, \pi_N)$ bezeichnet. Sie müssen folgende Bedingung erfüllen:

$$\sum_{i=1}^{N} \pi_i = 1.$$

Übergangswahrscheinlichkeiten: Die Wahrscheinlichkeiten aller möglichen Zustandsübergänge werden in Form einer $N \times N$-Matrix $A = (a_{ij})$ angegeben. Dabei kennzeichnet a_{ij} die Wahrscheinlichkeit, vom Zustand i in den Zustand j überzugehen. Es gilt die Nebenbedingung

$$\sum_{j=1}^{N} a_{ij} = 1, \qquad 1 \leq i \leq N.$$

Ausgaben: Jeder Zustand im HMM ist mit einer Ausgabefunktion verknüpft, gemäß der die Symbole der Sequenz generiert werden. Die Funktionen werden durch eine Menge von Parametern eindeutig festgelegt. Für die Wahrscheinlichkeitsdichten muss folgende Normierung erfüllt sein:

$$\int_{\infty}^{-\infty} f_j(x)\, dx = 1, \qquad 1 \leq j \leq N.$$

In der Mehrzahl der Anwendungen, bei denen kontinuierliche Modelle zum Einsatz kommen, werden eindimensionale Gauß-Dichten und auch konvexe Kombinationen von Gauß-Dichten verwendet. Wird der Mittelwert mit μ_{jm} und die Varianz mit σ_{jm}^2 bezeichnet, dann ergibt sich folgende Gestalt für die Ausgabefunktion im Zustand j:

$$f_j(x) = \sum_{m=1}^{M} c_{jm} \frac{1}{\sigma_{jm}\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu_{jm}}{\sigma_{jm}}\right)^2\right) \tag{2.1}$$

Im Sinne einer kompakten Schreibweise wird in dieser Arbeit, wie auch in der entsprechenden Literatur das Symbol λ zur Kennzeichnung der Gesamtheit aller Parameter eines Modells und das Symbol Λ zur Kennzeichnung der Parameter mehrerer Modelle verwendet. Häufig wird λ auch als Synonym für das eigentliche Hidden-Markov-Modell eingesetzt. Es sei darauf hingewiesen, dass keiner der Modell-Parameter von der Zeit abhängt, daher spricht man auch von einem zeithomogenen Modell.

Notation

In der Tabelle 2.1 sind einige relevante Größen zur HMM-Beschreibung zusammen mit der in der Arbeit verwendeten Notation aufgeführt. Im unteren Teil der Tabelle sind die beiden wichtigen Forward- und Backward Variablen α und β aufgeführt, deren Bedeutung im folgenden Abschnitt zu den HMM-Algorithmen genauer erklärt wird.

Unabhängigkeitsannahmen

In der Modellierung mit Hidden-Markov-Modellen werden die beiden folgenden Unabhängigkeitsannahmen gemacht. Sie sind wesentlich für die meisten der nachfolgenden Berechnungen und Algorithmen:

$$pr(q_t = i|q_{t-1}) = pr(q_t = i|q_{t-1}, q_{t-2}, q_{t-3}, \ldots), \tag{2.2}$$

$$p(O_t|q_t) = p(O_t|q_t, \lambda, O). \tag{2.3}$$

Die erste Annahme ist die bereits oben erwähnte Markov-Eigenschaft des Modells. Die zweite Annahme drückt aus, dass die Erzeugung eines Symbols O zum Zeitpunkt t lediglich vom vorliegenden Zustand q_t abhängt, alle übrigen Modellparameter haben keinen Einfluss und auch die

vorhergehenden und nachfolgenden Symbole der Sequenz spielen keine Rolle. Beide Annahmen sind Voraussetzung vieler Faktorisierungen von Wahrscheinlichkeiten bzw. Wahrscheinlichkeitsdichten, die im Verlauf dieser Arbeit durchgeführt werden.

Bedeutung	Symbol
Bedingte (diskrete) Wahrscheinlichkeit A, gegeben B	$pr(A\|B)$
Wahrscheinlichkeitsdichte einer Sequenz O bei gegebenem λ	$p(O\|\lambda)$
Likelihood von λ bei gegebenen Daten O	$L_O(\lambda)$
Ausgabesymbol zum Zeitpunkt t	O_t
Länge einer Sequenz	T
Sequenz	$O = O_1 O_2 \ldots O_T$
Teilsequenz von 1 bis t	$O_{(t)} = O_1 \ldots O_t$
Zustand zum Zeitpunkt t	q_t
Zustandspfad	$Q = q_1 q_2 \ldots q_T$
Anzahl Zustände	N
Anzahl Ausgabedichten	M
Anzahl Modelle	C
Übergangswahrscheinlichkeit von Zustand $i \to j$	$a_{ij} = pr(q_{t+1} = j \| q_t = i)$
Übergangsmatrix	$A = \{a_{ij}\}, \quad 1 \le i, j \le N$
Initialwahrscheinlichkeiten	$\pi = (\pi_1, \ldots, \pi_N)$
Ausgabedichte des i-ten Zustandes	$f_i(x)$
Gaußdichte mit Mittelwertvektor μ und Kovarianzmatrix Σ	$\mathcal{N}(x, \mu, \Sigma)$
Mischkoeffizient der Ausgabedichten	$c_{im} \to f_i(x) = \sum_m c_{im} \mathcal{N}(x, \mu_{im}, \Sigma_{im})$
Gesamtheit aller Parameter	$\lambda = (A, \pi, f(x) = (f_1(x), \ldots, f_N(x)))$
Parameter mehrerer Modelle	$\Lambda = (\lambda_1, \ldots, \lambda_C)$
Forward-Variable	$\alpha_t(i) = p(O_1 O_2 \ldots O_t, q_t = i \| \lambda)$
Backward-Variable	$\beta_t(i) = p(O_{t+1} O_{t+2} \ldots O_T \| q_t = i, \lambda)$

Tabelle 2.1: *Übersicht der verwendeten Symbole und Notation*

2.3 HMM-Algorithmen

In der Anwendung von Hidden-Markov-Modellen zur Analyse und Simulation von Bausparkollektiven sind zwei wesentliche Aufgaben zu erledigen:

1. Die Modellparameter müssen an die Sequenzen angepasst werden. Dies wird im Folgenden mit Training des Modells bezeichnet und die dabei verwendeten Sequenzen stellen die Trainingsdaten dar.

2. Zur Durchführung einer Simulation müssen bestehende Teilsequenzen geeignet verlängert und neue Sequenzen generiert werden.

Zur Bewältigung dieser beiden Aufgaben müssen die in den folgenden fünf Unterkapiteln vorgestellten HMM-Grundprobleme gelöst werden, die sich so auch in vielen anderen Anwendungsgebieten stellen. Hierzu wurden einige effiziente Algorithmen entwickelt, die in diesem Abschnitt vorgestellt werden sollen. Die Darstellung der ersten drei Algorithmen folgt weitgehend den Ausführungen in [32]. Die Algorithmen für diskrete und für kontinuierliche Modelle sind dabei vom Prinzip her recht ähnlich, wenngleich die expliziten Lösungsformeln natürlich unterschiedliche Gestalt haben. Da in der späteren Anwendung auf Bausparkollektive kontinuierliche Modelle verwendet werden sollen, werden die Algorithmen in diesem Kapitel auch anhand der kontinuierlichen Modelle hergeleitet. Die Vorgehensweise bei diskreten Modellen ist analog und findet sich z. B. in [23, 32].

2.3.1 Forward-Backward-Algorithmus

Problem 1 *Bestimme zu einer gegebenen Sequenz O der Länge T und einem gegebenem Modell λ die Wahrscheinlichkeitsdichte, dass die Sequenz von diesem Modell erzeugt wurde, d. h. berechne $p(O|\lambda)$.*

Die Berechnung von $p(O|\lambda)$ stellt den zentralen Baustein fast aller HMM-Algorithmen dar. Mit dem Forward-Backward-Algorithmus, der jetzt vorgestellt wird, existiert ein effizienter Algorithmus, der dieses Problem löst.
Der naive Ansatz zur Lösung dieses Problems besteht in einer Enumeration über alle Zustandspfade, die der vorgegebenen Sequenz zugrunde liegen könnten:

$$\begin{aligned} p(O|\lambda) &= \sum_{\text{erlaubte } Q} p(O, Q|\lambda) \\ &= \sum_{\text{erlaubte } Q} p(O|Q, \lambda) pr(Q|\lambda) \\ &= \sum_{\text{erlaubte } Q} \pi_{q_1} f_{q_1}(O_1) \prod_{t=2}^{T} a_{q_{t-1}q_t} f_{q_t}(O_t) \end{aligned} \tag{2.4}$$

Die Berechnung der Likelihood über Gleichung (2.4) hat eine Laufzeit von $O(TN^T)$, da im ungünstigsten Fall über N^T verschiedene Pfade enumeriert werden muss und jeder einzelne Pfad $O(T)$ Berechnungen erfordert. Diese Berechnung ist bereits für Instanzen moderater Größenordnung praktisch nicht durchführbar. Jedoch gibt es mit dem Forward-Backward-Algorithmus einen effizienten Algorithmus zur Berechnung von $p(O|\lambda)$, der nun vorgestellt werden soll.

Seien zunächst die Forward-Variable $\alpha_t(i)$ und die Backward-Variable $\beta_t(i)$ wie folgt definiert:

$$\alpha_t(i) := p(O_1 O_2 \ldots O_t, q_t = i|\lambda), \tag{2.5}$$

$$\beta_t(i) := p(O_{t+1} O_{t+2} \ldots O_T|q_t = i, \lambda). \tag{2.6}$$

Dann gilt für jedes t mit $1 \leq t \leq T$ und jedes i mit $1 \leq i \leq N$

$$p(O, q_t = i|\lambda) = \alpha_t(i)\beta_t(i), \tag{2.7}$$

denn

$$\begin{aligned} p(O, q_t = i|\lambda) &= p(O|q_t = i, \lambda)pr(q_t = i|\lambda) \\ &= p(O_1 \ldots O_t|q_t = i, \lambda)p(O_{t+1} \ldots O_T|q_t = i, \lambda)pr(q_t = i|\lambda) \\ &= p(O_1 \ldots O_t, q_t = i|\lambda)p(O_{t+1} \ldots O_T|q_t = i, \lambda) \\ &= \alpha_t(i)\beta_t(i). \end{aligned}$$

Das zweite Gleichheitszeichen gilt aufgrund der statistischen Unabhängigkeit einer Anfangs- und Endsequenz bei festgehaltenem Zustand zum Trennungszeitpunkt.
Mit (2.7) folgt unmittelbar

$$p(O|\lambda) = \sum_{i}^{N} \alpha_t(i)\beta_t(i). \tag{2.8}$$

Somit ist das Problem einer effizienten Berechnung von $p(O|\lambda)$ überführt in eine effiziente Berechnung der Variablen $\alpha_t(i)$ und $\beta_t(i)$. Zusätzlich gilt auch:

$$p(O|\lambda) = \sum_{i}^{N} \alpha_T(i). \tag{2.9}$$

Hieraus folgt, dass die Berechnung von $\beta_t(i)$ zur Lösung von Problem 1 gar nicht notwendig ist. Jedoch wird $\beta_t(i)$ zur Berechnung des Viterbi-Pfades (Problem 2 im Abschnitt 2.3.2) und zum Modelltraining (Problem 3 im Abschnitt 2.3.3) benötigt, und da die Vorgehensweise zur Berechnung von $\beta_t(i)$ stark der von $\alpha_t(i)$ ähnelt, wird sie bereits hier gezeigt.

Die Berechnung von $\alpha_t(i)$ ist über die folgende „Forward-Prozedur“ möglich; die Berechnung von $\beta_t(i)$ geschieht mit der nachfolgenden „Backward-Prozedur“.

Forward-Prozedur
1. Initialisierung:

$$\alpha_1(j) = \pi_j f_j(O_1), \qquad 1 \leq j \leq N.$$

2. Induktion über t:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i)a_{ij}\right] f_j(O_{t+1}), \quad 1 \leq j \leq N, \quad 1 \leq t \leq T. \tag{2.10}$$

Durch den Induktionsschritt (2.10) wird die Enumeration über alle Pfade vermieden. Für jeden Zeitschritt sind lediglich $O(N^2)$ Berechnungen notwendig, sodass sich eine Gesamtkomplexität von $O(N^2T)$ Rechenoperationen ergibt.

Das Vorgehen zur Berechnung von $\beta_t(i)$ ist ähnlich, die Iteration läuft aber in umgekehrter Zeitrichtung, beginnt also bei $t = T$ und endet bei $t = 0$.

Backward-Prozedur
1. Initialisierung:

$$\beta_T(j) := 1, \qquad 1 \le j \le N.$$

2. Induktion über t:

$$\beta_t(j) = \sum_{i=1}^{N} a_{ji} f_i(O_{t+1}) \beta_{t+1}(i), \quad 1 \le j \le N, \quad T-1 \ge t \ge 1. \tag{2.11}$$

Mit den gleichen Argumenten wie bei der Forward-Prozedur sieht man, dass die Berechnung von $\beta_t(i)$ $O(N^2T)$ Operationen erfordert.

Skalierter Forward-Backward-Algorithmus
Eine direkte Implementierung der Forward- und Backward-Prozedur nach den Gleichungen (2.10) und (2.11) kann zu Problemen führen, da in der Induktion besonders bei sehr langen Sequenzen sehr kleine Zahlen auftreten können, die eventuell den darstellbaren Zahlenbereich des Rechners verlassen. Ein Ausweg hieraus besteht in der Durchführung einer Skalierung, die die auftretenden Größen bei jeder Iteration sicher in den darstellbaren Zahlenbereich überführt. Beispielhaft wird diese Skalierung jetzt für die Forward-Prozedur demonstriert. Der Unterschied zu den ursprünglichen Berechnungen besteht darin, dass für jeden Zeitpunkt und jeden Zustand zwei Variablen eingeführt werden. Die Variable $\tilde{\alpha}_t(j)$ wird zunächst für alle Zustände berechnet, um dann zur Skalierung der Variablen $\hat{\alpha}_t(j)$ eingesetzt zu werden:

1. Initialisierung für $1 \le j \le N$:

$$\tilde{\alpha}_1(j) = \pi_j f_j(O_1), \tag{2.12}$$

$$\hat{\alpha}_1(j) = \tilde{\alpha}_1(j) / \sum_{i=1}^{N} \tilde{\alpha}_1(i). \tag{2.13}$$

2. Induktion über t für $1 \le j \le N$:

$$\tilde{\alpha}_{t+1}(j) = \left[\sum_{i=1}^{N} \hat{\alpha}_t(i) a_{ij} \right] f_j(O_{t+1}), \tag{2.14}$$

$$\begin{aligned} \hat{\alpha}_{t+1}(j) &= \tilde{\alpha}_{t+1}(j) / \sum_{i=1}^{N} \tilde{\alpha}_{t+1}(i) \\ &= c_{t+1} \tilde{\alpha}_{t+1}(j) \end{aligned} \tag{2.15}$$

mit dem Skalierungsfaktor c_t, der wie folgt definiert ist:

$$c_t := \frac{1}{\sum_{i=1}^{N} \tilde{\alpha}_t(i)}. \tag{2.16}$$

Die Skalierung der Backward-Prozedur wird in ganz analoger Weise durchgeführt, es werden die gleichen Skalierungsfaktoren nach Gleichung (2.16) wie bei der Forward-Prozedur eingesetzt (Details hierzu finden sich z. B. in [23]).

Per Induktion lässt sich zeigen [40], dass zwischen den ursprünglichen und den skalierten Variablen folgende Beziehungen gelten:

$$\hat{\alpha}_t(i) = \left(\prod_{\tau=1}^{t} c_\tau \right) \alpha_t(i), \tag{2.17}$$

$$\hat{\beta}_t(i) = \left(\prod_{\tau=t+1}^{T} c_\tau \right) \beta_t(i). \tag{2.18}$$

Da die Prozedur in der skalierten Version die ursprünglichen Variablen $\alpha_t(i)$ nicht berechnet, stellt sich das Problem, dass auch die Likelihood nach Gleichung (2.9) nicht mehr berechenbar ist. Dies wäre ja auch sehr verwunderlich, da die Likelihood auch außerhalb des darstellbaren Zahlenbereichs liegt. Der Ausweg liegt in der Berechnung des Logarithmus der Likelihood. Aus den Gleichungen (2.15) und (2.16) folgt

$$\sum_{i=1}^{N} \hat{\alpha}_T(i) = 1. \tag{2.19}$$

Daraus folgt mit Gleichung (2.17):

$$1 = \sum_{i=1}^{N} \hat{\alpha}_T(i) = \sum_{i=1}^{N} \left(\prod_{\tau=1}^{T} c_\tau \right) \alpha_T(i).$$

Mit Gleichung (2.9) gilt somit

$$p(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i) = 1 / \prod_{\tau=1}^{T} c_\tau. \tag{2.20}$$

Das Produkt $\prod_{\tau=1}^{T} c_\tau$ ist wiederum sehr klein, aber die Summe lässt sich auf jeden Fall berechnen, sodass schließlich gilt:

$$\log(p(O|\lambda)) = -\sum_{\tau=1}^{T} \log(c_\tau). \tag{2.21}$$

In der Implementierung wird durchgehend mit dem natürlichen Logarithmus gearbeitet. Aus theoretischer Sicht ist es jedoch gleichgültig, welche Basis verwendet wird.

2.3.2 Viterbi-Algorithmus

Problem 2 *Bestimme bei einer gegebenen Sequenz O und einem gegebenen Modell λ den Zustandspfad mit der größten Wahrscheinlichkeit zur Erzeugung dieser Sequenz.*

In der Modellierung der Bauspardaten mit Hidden-Markov-Modellen taucht diese Problematik bei der Verlängerung von Teilsequenzen auf. Details hierzu finden sich im Abschnitt 6.2. Die

Betonung bei dieser Problemstellung liegt in der Maximierung der Wahrscheinlichkeit bezogen auf die Gesamtsequenz. So ist der nahe liegende Versuch, zu jedem Zeitpunkt gerade jenen Zustand auszuwählen, der zur gegebenen Anfangssequenz die größte Wahrscheinlichkeit besitzt, in der Regel keine Lösung, da hierdurch ein Pfad entstehen könnte, der mit der vorgegebenen Übergangsmatrix nicht verträglich ist, weil er verbotene Zustandswechsel enthält.

Der Viterbi-Algorithmus zur Lösung von Problem 2 wird hier direkt in der skalierten Version angegeben. Er arbeitet mit der Methode der dynamischen Programmierung und der resultierende Zustandspfad wird mit Viterbi-Pfad bezeichnet. Dazu wird zunächst folgende Größe definiert:

$$\delta_t(i) := \max_{q_1,\ldots,q_{t-1}} \log(L[q_1,\ldots,q_t = i, O_1,\ldots,O_t|\lambda]).$$

$\delta_t(i)$ ist die Log-Likelihood der Teil-Sequenz und des maximalen Teil-Pfades, der den ersten t Ausgabesymbolen Rechnung trägt und zum Zeitpunkt t im Zustand i ist. Es gilt folgende Induktionsformel:

$$\delta_{t+1}(j) = \max_i \left[\delta_t(i) + \log(a_{ij})\right] + \log(f_j(O_{t+1})).$$

Viterbi-Algorithmus

1. Initialisierung:

$$\begin{aligned} \delta_1(j) &= \log(\pi_i f_j(O_1)), \qquad 1 \le j \le N, \\ \psi_1(j) &= 0. \end{aligned}$$

2. Rekursion:

$$\begin{aligned} \delta_t(j) &= \max_{1\le i\le N} \left[\delta_{t-1}(i) + \log(a_{ij})\right] + \log(f_j(O_t)), && 2 \le t \le T, \\ & && 1 \le j \le N, \\ \psi_t(j) &= \arg\max_{1\le i\le N}[\delta_{t-1}(i) + \log(a_{ij})], && 2 \le t \le T, \\ & && 1 \le j \le N. \end{aligned}$$

3. Terminierung:

$$\begin{aligned} p^\star &= \max_{1\le i\le N}[\delta_T(i)], \\ q_T^\star &= \arg\max_{1\le i\le N}[\delta_T(i)]. \end{aligned}$$

4. Pfad-Backtracking:

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

Der wesentliche Unterschied zum Forward-Backward-Algorithmus besteht darin, dass hier im Rekursionsschritt von allen möglichen Zustandsübergängen nur derjenige berücksichtigt wird, der den größten Beitrag liefert, während beim Forward-Backward-Algorithmus die Summe über alle möglichen Übergänge gebildet wird. Zusätzlich wird der damit korrespondierende Ausgangszustand in einer Matrix gespeichert. Diese Matrix wird im Backtracking-Schritt zur Rekonstruktion des Viterbi-Pfades genutzt.

2.3.3 Baum-Welch-Algorithmus

Problem 3 (Modelltraining) *Bestimme zu einer vorgegebenen Menge von Sequenzen* $O = (O^1, \ldots, O^K)$ *und zu gegebener Modelltopologie die Parameter* $\hat{\lambda}$*, die die Gesamtdichte*

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

maximieren. $\hat{\lambda}$ *wird auch Maximum Likelihood Schätzer genannt.*

Der Baum-Welch-Algorithmus zur Lösung dieses Problems wurde bereits in den späten sechziger Jahren von Baum und anderen entwickelt [2,3]. Einige Jahre später (1977) haben Dempster, Laird und Rubin ein neues, sehr allgemeines Schema zur Maximum-Likelihood-Schätzung von Parametern vorgestellt [8]. Dieser so genannte EM-Algorithmus stellt eine Verallgemeinerung des Baum-Welch-Algorithmus dar. Die Formeln des Baum-Welch-Algorithmus (im Folgenden auch Reestimierungsformeln genannt) werden in diesem Abschnitt zunächst durch Plausibilitätsbetrachtungen begründet; im Abschnitt 2.5 wird dann das allgemeinere Schema des EM-Algorithmus einschließlich eines Konvergenzbeweises vorgestellt. Zunächst werden die unskalierten Formeln angegeben, da sie eine anschauliche Interpretation besitzen, anschließend wird gezeigt, wie sie durch die skalierten Variablen dargestellt werden können.

Beim Baum-Welch-Algorithmus handelt es sich um einen iterativen Algorithmus, der ausgehend von einer Initialbelegung aller Parameter neue Parameter findet, die zumindest nicht schlechter sondern in der Regel besser sind als die bisherigen. In jeder Iteration werden auf Basis der alten Parameter neue Parameter berechnet, die im Folgeschritt die Rolle der alten Größen übernehmen. Dieses Verfahren wird so lange fortgesetzt, bis die Verbesserung der Likelihood unterhalb einer vorzugebenden Schranke gefallen ist.

Der Baum-Welch-Algorithmus findet in der Regel nur eine lokal optimale Lösung. Die Parameter, die der Algorithmus liefert, hängen also von der Wahl der Initialparameter ab. Daher ist es in praktischen Anwendungen ratsam, mehrere Durchläufe des Verfahrens mit unterschiedlichen Initialisierungen zu betrachten und die Lösung mit maximaler Likelihood zu speichern. Bislang ist kein effizientes Verfahren bekannt, das mit Sicherheit eine global optimale Lösung von Problem 3 liefert.

Vorgaben
Es seien K statistisch voneinander unabhän-gige Sequenzen $O = (O^1, \ldots, O^K)$ mit den individuellen Längen $(T^1, \ldots, T^K)$ vorgegeben. Gesucht wird der Satz von Parametern $\hat{\lambda}$, der die folgende Likelihood maximiert:

$$L_O(\lambda) := p(O^1, \ldots, O^K|\lambda) = \prod_{k=1}^{K} p(O^k|\lambda). \tag{2.22}$$

Dies ist äquivalent zur Maximierung von:

$$\log\left(p(O^1, \ldots, O^K|\lambda)\right) = \sum_{k=1}^{K} \log\left(p(O^k|\lambda)\right). \tag{2.23}$$

Als Dichtefunktion für die Ausgaben im Zustand $i,\ 1 \le i \le N$ wird eine Mischung von multivariaten Normaldichten angenommen:

$$\begin{aligned} f_i(x) &= \sum_{m=1}^{M} c_{im}\mathcal{N}(x, \mu_{im}, \Sigma_{im}) \\ &=: \sum_{m=1}^{M} c_{im} f_{im}(x), \end{aligned}$$

mit

$$\mathcal{N}(x, \mu, \Sigma) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)'\Sigma^{-1}(x-\mu)\right) \tag{2.24}$$

und

$$\sum_{m=1}^{M} c_{im} = 1, \quad c_{im} \ge 0, \quad 1 \le i \le N,$$

wobei d die Dimension von x, Σ die Kovarianzmatrix und $(\cdot)'$ den transponierten Vektor kennzeichnen. Es gibt noch allgemeinere Ansätze in der HMM-Literatur, die als Dichtefunktion eine Überlagerung multivariater, strikt log-konkaver und/oder elliptisch symmetrischer Funktionen verwenden [18, 28]. In dieser Arbeit werden im Folgenden jedoch ausschließlich multivariate und univariate Gauß-Dichten betrachtet.

Als Abbruchkriterien müssen noch Schranken ε und MI vorgegeben werden: Wenn die Differenz der Zielfunktionswerte zweier aufeinander folgender Iterationen kleiner als ε ist oder wenn die maximale Zahl von Iterationen MI erreicht ist, stoppt der Algorithmus.

Hilfsvariablen

Die Darstellung des Baum-Welch-Algorithmus durch die Angabe von Update-Formeln für die Modellparameter (s. Punkt „Iteration“) wird deutlich kompakter, wenn für jede Sequenz O^k weitere Hilfsvariablen definiert werden. Soweit nicht anders angegeben, gilt für die nachfolgenden Formeln dieses Abschnitts $1 \le i, j \le N$, $1 \le t \le T$ und $1 \le m \le M$.
Die Hilfsvariablen seien wie folgt definiert:

$$\begin{aligned} \gamma_t^k(i) &:= pr(q_t = i | O^k, \lambda), \\ \xi_t^k(i,j) &:= pr(q_t = i, q_{t+1} = j | O^k, \lambda), \\ \zeta_t^k(i,m) &:= pr(q_t = i, m | O^k, \lambda). \end{aligned}$$

Bei gegebener Sequenz O^k und λ gibt $\gamma_t^k(i)$ die Wahrscheinlichkeit an, zum Zeitpunkt t im Zustand i zu sein, während $\xi_t^k(i,j)$ die Verbundwahrscheinlichkeit kennzeichnet, zum Zeitpunkt t im Zustand i und zum Zeitpunkt $t+1$ im Zustand j zu sein. Mit $\zeta_t^k(i,m)$ wird die gemeinsame Wahrscheinlichkeit vom Zustand i und der Mischkomponente m zum Zeitpunkt t bezeichet. Es gilt die Beziehung:

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

$\gamma_t^k(i)$ kann mit den durch Gleichung (2.5) und Gleichung (2.6) definierten Forward- und Backward-Variablen $\alpha_t^k(i)$, $\beta_t^k(i)$ unter Verwendung von (2.7) und (2.8) wie folgt berechnet werden (der zusätzliche Index *k* bei α und β bedeutet, dass sich die Variablen auf die Sequenz O^k beziehen):

$$\begin{aligned}\gamma_t^k(i) &= \frac{p(O^k, q_t = i|\lambda)}{p(O^k|\lambda)} \\ &= \frac{\alpha_t^k(i)\beta_t^k(i)}{\sum_{j=1}^N \alpha_t^k(j)\beta_t^k(j)}.\end{aligned} \tag{2.25}$$

Für $\xi_t^k(i,j)$ gilt analog ($1 \le t \le T-1$):

$$\begin{aligned}\xi_t^k(i,j) &= \frac{p(O^k, q_t = i, q_{t+1} = j|\lambda)}{p(O^k|\lambda)} \\ &= \frac{\alpha_t^k(i)a_{ij}f_j(O_{t+1}^k)\beta_{t+1}^k(j)}{\sum_{l=1}^N \alpha_t^k(l)\beta_t^k(l)}\end{aligned} \tag{2.26}$$

und für $\zeta_t^k(i,m)$ gilt schließlich:

$$\zeta_t^k(j,m) = \frac{p(O^k, q_t = j, m|\lambda)}{p(O^k|\lambda)}.$$

Auch dieser Ausdruck kann durch die Forward-Backward-Variablen α, β ausgedrückt werden. Hierbei muss dann aber zwischen $t = 1$ und $t > 1$ unterschieden werden:

$$\begin{aligned}\zeta_1^k(j,m) &= \frac{\pi_j c_{jm} f_{jm}(O_1^k)\beta_1^k(j)}{\sum_{i=1}^N \alpha_1^k(i)\beta_1^k(i)} \\ \zeta_t^k(j,m) &= \frac{\sum_{i=1}^N (\alpha_{t-1}^k(i)a_{ij})c_{jm} f_{jm}(O_t^k)\beta_t^k(j)}{\sum_{i=1}^N \alpha_t^k(i)\beta_t^k(i)}, \quad 2 \le t \le T.\end{aligned} \tag{2.27}$$

Baum-Welch-Algorithmus

Initialisierungen
Da der Baum-Welch-Algorithmus ein iterativer Algorithmus ist, der aus bestehenden Parametern neue Parameter berechnet, ist es notwendig, alle Modellparameter zu initialisieren. Dies ist ein wichtiger Aspekt bei der praktischen Anwendung des Algorithmus, da die erhaltene Lösung von der gewählten Initialisierung abhängt. Einzelheiten dazu, was bei der Initialbelegung der Parameter zu beachten ist und wie sich verschiedene Initialisierungen auswirken, werden im Abschnitt 5.2 erläutert.

Reestimierungsformeln
Der Kern des Baum-Welch-Algorithmus besteht in der Wiederholung der beiden folgenden Berechnungen bis zum Erreichen der Konvergenz bzw. bis zum Erreichen der maximalen Iterationszahl:

E-Schritt: Berechnung der Hilfsvariablen $\gamma_t^k(i)$ (Gleichung (2.25)), $\xi_t^k(i,j)$ (Gleichung (2.26)) und $\zeta_t^k(j,m)$ (Gleichung (2.27)) basierend auf dem aktuellen Parameter λ für alle i, j, k, t.

M-Schritt: Berechnung des neuen Parameters $\bar{\lambda}$ unter Zugrundelegung des aktuellen Parameters λ und der im E-Schritt berechneten Hilfsvariablen mittels folgender Reestimierungsformeln für alle i, j, m:

$$\bar{\pi}_i = \frac{\sum_{k=1}^{K} \alpha_1^k(i)\beta_1^k(i)}{\sum_{k=1}^{K}\sum_{j=1}^{N} \alpha_1^k(j)\beta_1^k(j)} \tag{2.28}$$

$$\bar{a}_{ij} = \frac{\sum_{k=1}^{K}\sum_{t=1}^{T^k-1} \xi_t^k(i,j)}{\sum_{k=1}^{K}\sum_{t=1}^{T^k-1} \gamma_t^k(i)} \tag{2.29}$$

$$\bar{c}_{jm} = \frac{\sum_{k=1}^{K}\sum_{t=1}^{T^k} \zeta_t^k(j,m)}{\sum_{k=1}^{K}\sum_{t=1}^{T^k} \gamma_t^k(i)} \tag{2.30}$$

$$\bar{\mu}_{jm} = \frac{\sum_{k=1}^{K}\sum_{t=1}^{T^k} \zeta_t^k(j,m) O_t^k}{\sum_{k=1}^{K}\sum_{t=1}^{T^k} \zeta_t^k(j,m)} \tag{2.31}$$

$$\bar{\Sigma}_{jm} = \frac{\sum_{k=1}^{K}\sum_{t=1}^{T^k} \zeta_t^k(j,m)(O_t^k - \mu_{jm})(O_t^k - \mu_{jm})'}{\sum_{k=1}^{K}\sum_{t=1}^{T^k} \zeta_t^k(j,m)}. \tag{2.32}$$

Setzen des neuen Parameters $\bar{\lambda}$ auf den aktuellen Parameter λ.

Die Bezeichnungen „E-Schritt„ und „M-Schritt“ werden bei der Beschreibung des EM-Algorithmus im Abschnitt 2.5 wieder aufgegriffen und erläutert. Jede der obigen Reestimierungsformeln besitzt eine schöne anschauliche Interpretation. Beispielhaft seien Nenner und Zähler für die Reestimierung der Übergangsparameter a_{ij} genauer betrachtet (für die übrigen Reestimierungsformeln können analoge Interpretationen angegeben werden). Der Zähler

$$\sum_{k=1}^{K}\sum_{t=1}^{T-1} \xi_t^k(i,j)$$

kann als erwartete Häufigkeit der Übergänge vom Zustand i in den Zustand j bei gegebenem λ und gegebenen Sequenzen $(O^1, \ldots, O^K)$ interpretiert werden. Es ist eine erwartete Häufigkeit, da natürlich die „echte“ Häufigkeit verborgen bleibt. Der Nenner

$$\sum_{k=1}^{K}\sum_{t=1}^{T^k-1} \gamma_t^k(i)$$

repräsentiert dann die erwartete Häufigkeit aller Übergänge aus dem Zustand i heraus, unabhängig davon, in welchen Zustand gewechselt wird. Es ist plausibel, als Schätzer für a_{ij} den Quotienten aus der erwarteten Häufigkeit der Übergänge vom Zustand i in den Zustand j und der erwarteten Häufigkeit aller Übergänge aus dem Zustand i heraus zu wählen. Die mathematische Begründung der Baum-Welch Reestimierungsformeln wird im Abschnitt 2.5 geliefert. Dort wird

am Beispiel der Reestimierungsformel für π gezeigt, wie die Formeln mit Hilfe des allgemeinen EM-Schemas hergeleitet werden können. Es wird auch die Konvergenz des EM-Schemas bewiesen, woraus sich unmittelbar die Konvergenz des Baum-Welch-Algorithmus ergibt, da letzterer die Anwendung des EM-Algorithmus auf Hidden-Markov-Modelle ist.

Reestimierung und Skalierung
Die Reestimierungsformeln können nicht direkt implementiert werden, da sie über die drei Hilfsvariablen noch die unskalierten Variablen α und β enthalten. Am Beispiel von $\gamma_t^k(i)$ soll nun gezeigt werden, wie sich dieses Problem auflösen lässt.

In den Reestimierungsformeln für a_{ij} und c_{jm} taucht $\gamma_t^k(i)$ ausschließlich als Summe über t auf. Unter Verwendung der Gleichungen (2.17), (2.18) und (2.20) gilt für diese Summe:

$$\begin{aligned}
\sum_{t=1}^{T^k-1} \gamma_t^k(i) &= \sum_{t=1}^{T^k-1} \frac{1}{p(O^k|\lambda)} \alpha_t^k(i)\beta_t^k(i) \\
&= \sum_{t=1}^{T^k-1} \left(\prod_{\tau=1}^{T^k} c_\tau \right) \alpha_t^k(i)\beta_t^k(i) \\
&= \sum_{t=1}^{T^k-1} \left(\prod_{\tau=1}^{t} c_\tau \right) \alpha_t^k(i) \left(\prod_{\tau=t+1}^{T^k} c_\tau \right) \beta_t^k(i) \\
&= \sum_{t=1}^{T^k-1} \hat{\alpha}_t^k(i)\hat{\beta}_t^k(i). \qquad (2.33)
\end{aligned}$$

Ähnlich wie für $\gamma_t^k(i)$ können auch die Summationen über die beiden anderen Hilfsvariablen in den Reestimierungsformeln sehr elegant durch die skalierten $\hat{\alpha}$ und $\hat{\beta}$ ausgedrückt werden. Die sich ergebenden Formeln haben sogar eine einfachere Form, da der ursprüngliche Nenner $p(O^k|\lambda)$ verschwindet. Eine ausführliche Zusammenstellung aller skalierten Reestimierungsformeln findet sich in [23].

2.3.4 Klassifikation von Sequenzen

Problem 4 (Klassifikation) *Bestimme die Klasse einer Sequenz, d. h. bestimme von einer gegebenen Menge von Hidden-Markov-Modellen das Modell, von dem die Sequenz mit der größten Wahrscheinlichkeit generiert wurde. Die Modelle sind hierbei vorgegeben und werden nicht verändert.*

Dieser Problemstellung liegt die Annahme zugrunde, dass es eine endliche Menge von Klassen gibt, denen eine gegebene Sequenz angehören kann. Diese Klassen werden durch eine entsprechende Anzahl von Hidden-Markov-Modellen repräsentiert. Gesucht ist ein Klassifikator, der für jede Sequenz entscheiden kann, zu welchem der möglichen Modelle sie gehört.

In verschiedenen Anwendungsgebieten werden Klassifikatoren benötigt: So bildet in der Handschriftenerkennung jeder Buchstabe eine Klasse. In der automatischen Spracherkennung stellen so genannte Phoneme eine natürliche Klasseneinteilung dar. In beiden Fällen sind die Klassenzugehörigkeiten der Trainingsdaten in der Regel bekannt. Dies hat den großen Vorteil, dass die Qualität eines Klassifizierungsalgorithmus direkt beurteilt werden kann, indem die Zahl der falsch klassifizierten Sequenzen der Zahl der korrekten Beispiele gegenüber gestellt wird.

In der Anwendung dieser Arbeit, der Modellierung von Bausparkollektiven, ist der Klassenbegriff eher abstrakter Natur, es ist die Vorstellung der Existenz bestimmter prototypischer Verhaltensweisen von Bausparern. Hier ist die Klassenzugehörigkeit einer Sequenz nicht bekannt. Ein Vergleich der Güte verschiedener Klassifizierungen geschieht anhand einer geeignet zu definierenden Zielfunktion.

Das Problem, Sequenzen zu klassifizieren, taucht in dieser Arbeit an drei Stellen auf:

HMM-Clusterung: Bei der Clusterung mit Hidden-Markov-Modellen (Abschnitt 2.3.5) muss in jeder Iteration für alle Sequenzen entschieden werden, zu welchem der zur Auswahl stehenden Modelle sie jeweils am besten passen.

Sequenzverlängerung: In Paragraph 6.2, wo es um die Verlängerung von Teilsequenzen geht, stellt sich das gleiche Problem: Welches der möglichen Hidden-Markov-Modelle soll als Generator der Endsequenz verwendet werden?

Mischmodelle: Bei der Clusterung komplexer Daten mit einem Mischmodellansatz (s. Kapitel 3) geht die Wahrscheinlichkeit der Modelle als Gewichtung direkt in die Schätzung der Modellparameter ein. Dies ist eine Erweiterung der Klassifizierungsaufgabe.

Wenn mit λ_i die Parameter des i-ten Modells und mit Λ die Gesamtheit der betrachteten Modellparameter bezeichnet werden, lassen sich die zwei in dieser Arbeit verwendeten Funktionen zur Klassifizierung wie folgt angeben:

1. MD (= maximale Dichte) Klassifizierung: Wähle das Modell i, für das der Logarithmus der Wahrscheinlichkeitsdichte $\log(p(O|\Lambda, i) \equiv \log(p(O|\lambda_i))$ maximal ist.
2. MAW (= maximale a posteriori Wahrscheinlichkeit) Klassifizierung: Wähle das Modell i, für das der Logarithmus der a posteriori Wahrscheinlichkeit $\log(pr(i|O, \Lambda))$ maximal ist.

Bei der HMM-Clusterung wird ausschließlich der MD-Klassifikator eingesetzt, da der dort verwendete Baum-Welch-Algorithmus dies erforderlich macht. Details zu diesem Punkt finden sich im folgenden Abschnitt 2.3.5.

Es ist sinnvoll, als Klassifizierungsfunktion den Logarithmus der Wahrscheinlichkeitsdichten zu wählen, da die Wahrscheinlichkeitsdichte $p(O|\lambda_i)$ so klein werden kann, dass die Genauigkeit der Rechnerdarstellung nicht ausreichend ist. Aus Sicht der Klassifizierung ist es äquivalent, ob $p(\cdot)$ oder $\log(p(\cdot))$ verwendet wird, da der Logarithmus eine streng monotone Funktion ist.

Der Logarithmus der a posteriori Wahrscheinlichkeit eines Modells kann aus der Wahrscheinlichkeitsdichte einer Sequenz über die Bayes-Formel berechnet werden:

$$\begin{aligned} \log(pr(i|O,\Lambda)) &= \log\left(\frac{pr(i|\Lambda)p(O|\Lambda,i)}{p(O|\Lambda)}\right) \\ &= \log(pr(i|\Lambda)) + \log(p(O|\Lambda,i)) - \log(p(O|\Lambda)). \end{aligned} \tag{2.34}$$

Die Wahrscheinlichkeitsdichte $p(O|\Lambda)$ ist eine Konstante und muss daher bei der Klassifizierung nicht berücksichtigt werden. Somit liefern die beiden Klassifikatoren bei gegebener Sequenz O das Modell i als wahrscheinlichstes Modell, für das jeweils die folgende Bedingung gilt:

$$\textbf{MD:} \quad \text{classify}(O) := \arg\max_i(\log(p(O|\lambda_i))), \tag{2.35}$$

$$\textbf{MAW:} \quad \text{classify}(O) := \arg\max_i \left(\log(p(O|\Lambda, i)) + \log(pr(i|\Lambda))\right). \tag{2.36}$$

Die Berechnung von $\log(p(O|\lambda_i))$ geschieht mit dem Forward-Algorithmus gegeben durch Gleichung (2.21). Die Größe $pr(i|\Lambda)$ ist die Wahrscheinlichkeit des Modells i mit den Parametern λ_i im Vergleich zu den übrigen betrachteten Modellen, die sich ohne Berücksichtigung von Daten ergibt (a priori Wahrscheinlichkeit). Wenn jedes Modell als gleich wahrscheinlich angesehen werden kann, sind die beiden Klassifikatoren äquivalent.

Die MAW-Klassifizierung bietet gegenüber der MD-Klassifizierung Vorteile bei der Behandlung kurzer Teilsequenzen, die als unvollständige Daten interpretiert werden können. Die Klassifizierung unvollständiger Daten ist naturgemäß mit einer größeren Unsicherheit verbunden als die Klassifizierung der vollständigen Sequenzen. Diese Unsicherheit wird bei der MAW-Klassifizierung dadurch verringert, dass a priori Wahrscheinlichkeiten der zur Auswahl stehenden Modelle in die Klassifizierung einfließen.

2.3.5 HMM-Clusterung

Problem 5 (HMM-Clusterung) *Bestimme zu einer vorgegebenen Anzahl von Hidden-Markov-Modellen mit fester Topologie und einer gegebenen Menge von Sequenzen die Modellparameter und die Zuordnung von Sequenzen zu den Modellen, die eine noch genauer zu spezifizierende Zielfunktion maximieren.*

Im vorigen Abschnitt zur Klassifizierung von Sequenzen wurde von einer Menge fester Modelle ausgegangen; sämtliche Modellparameter waren also vorgegeben. Die HMM-Clusterung, um die es in diesem Abschnitt gehen soll, stellt eine Erweiterung dahingehend dar, dass optimale Hidden-Markov-Modelle zur Beschreibung der gegebenen Sequenzmenge gesucht werden. Mit dieser Sichtweise handelt es sich um eine Kombination der Probleme 3 und 4, dem Training von Hidden-Markov-Modellen und der Klassifizierung von Sequenzen. Bei der HMM-Clusterung werden jeweils die Sequenzen der gleichen Klasse zum Training des mit dieser Klasse korrespondierenden Hidden-Markov-Modells eingesetzt. Diese Menge von Sequenzen wird hier auch als Cluster bezeichnet.

Der Algorithmus zur Lösung von Problem 5 ist ein zentraler Bestandteil vom Bausparmodell HMBM (Kapitel 4) und wurde in [23] im Detail vorgestellt. Der HMM-Clusteralgorithmus besitzt starke Analogien zu einem bekannten geometrischen Clusterverfahren, dem so genannten k-means-Algorithmus [13]. Eine gute Einführung zu diesem und zu anderen Clusterverfahren bietet [17].

Vorgaben und Zielfunktion
Gegeben sind K Sequenzen unterschiedlicher Länge und C Hidden-Markov-Modelle mit fester

Topologie:

$$\begin{aligned} O &= (O^1, \ldots, O^K), \\ \Lambda &= (\lambda_1, \ldots, \lambda_C). \end{aligned}$$

Im Folgenden bezeichnet $\mathcal{C}_i$ das i-te Cluster, also die Menge von Sequenzen für die gilt:

$$\mathcal{C}_i := \{O^k | \text{ classify}(O^k) = i\}$$

und $|\mathcal{C}_i|$ sei die Anzahl der Sequenzen in Cluster i.

Die zu maximierende Zielfunktion für den MD-Klassifikator $Z_{MD}(O, \Lambda)$ hat folgende Gestalt:

$$\begin{aligned} Z_{MD}(O, \Lambda) &:= \log \left(\prod_{i=1}^{C} \prod_{O^k \in \mathcal{C}_i} p(O^k | \lambda_i) \right) \\ &= \sum_{i=1}^{C} \sum_{O^k \in \mathcal{C}_i} \log(p(O^k | \lambda_i)). \end{aligned} \tag{2.37}$$

Die Maximierung von (2.37) beinhaltet die folgenden zwei Teilprobleme:

1. Es muß eine optimale Partition der Sequenzen ermittelt werden. Die mit der Zahl der Sequenzen exponentiell steigende Zahl von Möglichkeiten, Sequenzen zu Gruppen zusammenzufassen, verhindert bei für die Praxis relevanten Größenordnungen die Durchführung eines enumerativen Verfahrens.
2. Bei gegebener Partition und damit bei fest definierten Trainingsmengen müssen die optimalen Modellparameter $\hat{\lambda}$ zur Beschreibung der Daten bestimmt werden.

Da im Allgemeinen keine exakte Lösung dieser beiden Probleme gefunden werden kann, wird man sich damit zufrieden geben müssen, einen Algorithmus wie den nachfolgend beschriebenen HMM-Clusteralgorithmus einzusetzen, der eine zumindest lokal optimale Lösung finden kann.

HMM-Clusteralgorithmus
Der HMM-Clusteralgorithmus gliedert sich in zwei Abschnitte. Nach der Initialisierung werden die nachfolgenden Berechnungsschritte bis zum Erreichen einer Abbruchbedingung iteriert.
Initialisierung:
Folgende Initialisierungen müssen vorgenommen werden:

- Belegung aller Modelle λ_i mit geeigneten Initialparametern.
- (optional) Vorgabe einer Sequenzpartition. Dies ersetzt die Klassifikation im ersten Iterationsschritt.

Es gibt verschiedene Möglichkeiten, diese Initialisierungen vorzunehmen. Details hierzu werden im Abschnitt 5.2 erläutert.

Iteration:
Wiederhole die folgenden Schritte bis zum Erreichen der Abbruchbedingung:

1. (a) Bestimme für alle Modelle λ_i die Cluster $\mathcal{C}_i$ durch Klassifikation aller Sequenzen O^k: classify(O^k).
   (b) Verhindere ggf. durch geeignetes Tauschen von Sequenzen, dass Modelle entstehen, denen keine Sequenzen zugeordnet sind (im Folgenden auch „leere Modelle“ genannt).
2. Trainiere jedes Modell mit dem Baum-Welch-Algorithmus basierend auf den Sequenzen des Clusters. Verwende die jeweiligen trainierten Parameter aus der vorangegangenen Iteration (bei der ersten Iteration die Initialparameter) als Ausgangspunkt des Trainings.
3. Berechne den neuen Zielfunktionswert mittels Gleichung (2.37).
4. Falls die maximale Zahl von Iterationen erreicht ist oder die Verbesserung des Zielfunktionswertes unter dem vorgegebenen Wert liegt, breche ab; anderfalls gehe zu Schritt eins.

Die Schritte einer Iteration im HMM-Clusteralgorithmus sind in Abbildung 2.2 veranschaulicht.

Konvergenz
In [23] wird bewiesen, dass der oben beschriebene Clusteralgorithmus gegen einen festen Wert der Zielfunktion konvergiert, wenn kein Austausch von Sequenzen zur Verhinderung leerer Modelle durchgeführt wird. Der Beweis basiert darauf, dass

1. die Startlösung einen endlichen Zielfunktionswert besitzt,
2. die Zielfunktion nach oben beschränkt ist,
3. keiner der Iterationsschritte zu einer Verschlechterung der Zielfunktion führen kann.

Der Wechsel von Sequenzen zur Verhinderung leerer Modelle kann dazu führen, dass der dritte Punkt verletzt ist. Da das Auftreten leerer Modelle nur sehr selten vorkommt, hat diese Einschränkung aus praktischer Sicht wenig Bedeutung.

Im Abschnitt 2.3.4 wurde neben dem MD-Klassifikator auch der MAW-Klassifikator vorgestellt. Kann auch diese Möglichkeit der Zuordnung von Sequenzen zu Modellen im HMM-Clusteralgorithmus genutzt werden? Dies ist nicht sinnvoll, da die Konvergenz des Verfahrens nicht gesichert ist. Die Clusterung mit dem MAW-Klassifikator hätte folgende Zielfunktion zu maximieren:

$$\begin{aligned} Z_{MAW}(O,\Lambda) &:= \log\left(\prod_{i=1}^{C}\prod_{O^k\in\mathcal{C}_i} pr(i|O^k,\Lambda)\right) \\ &= \sum_{i=1}^{C}\sum_{O^k\in\mathcal{C}_i}\log(pr(i|O^k,\lambda)). \end{aligned} \tag{2.38}$$

Unter Verwendung der Gleichung (2.34) gilt:

$$Z_{MAW}(O,\Lambda) = \sum_{i=1}^{C}\sum_{O^k\in\mathcal{C}_i}\left[\log(pr(i|\Lambda)) + \log(p(O^k|\lambda_i)) - \log(p(O^k|\Lambda))\right].$$

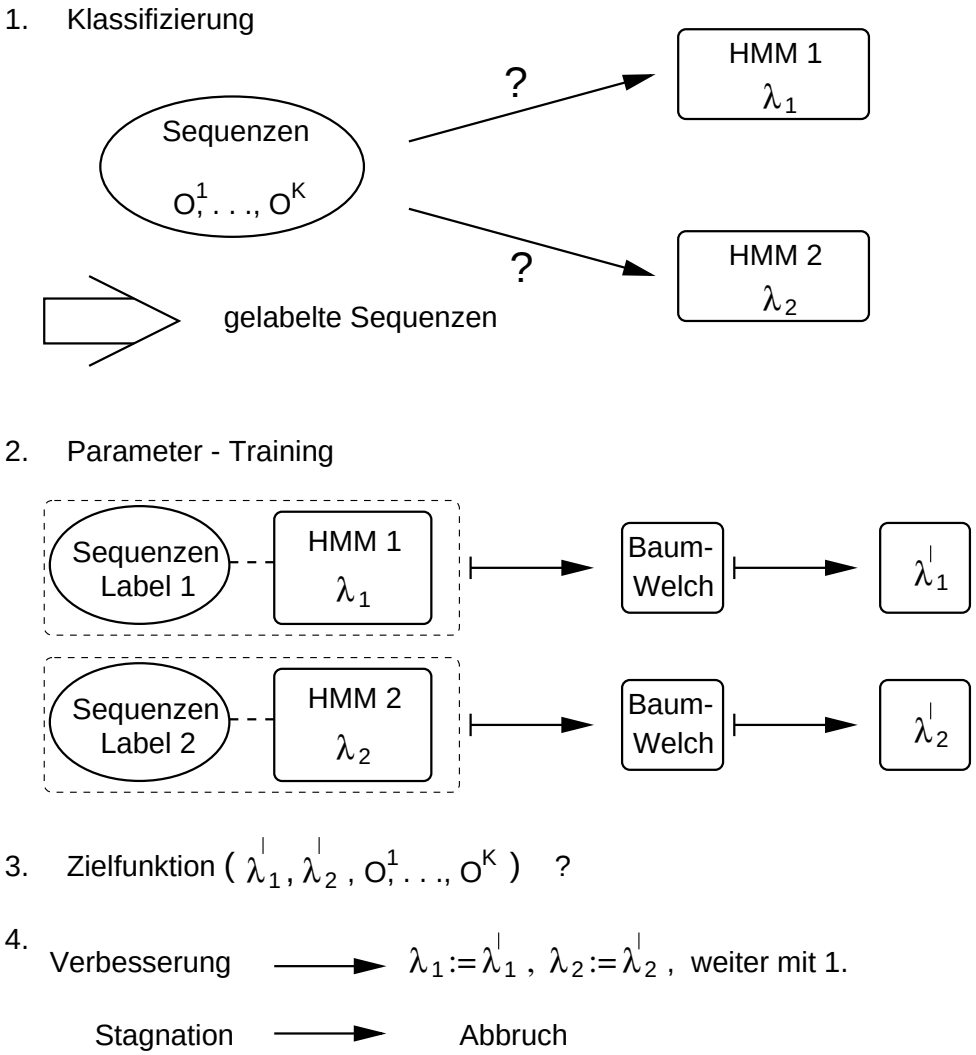

Abbildung 2.2: *Eine Iteration im HMM-Clusteralgorithmus am einfachen Beispiel von zwei Hidden-Markov-Modellen*

Die Klassifikation verbessert oder beläßt diese Zielfunktion nach Voraussetzung gleich. Der Baum-Welch-Algorithmus kann jedoch zu einer Verschlechterung führen, da es möglich ist, dass der letzte Ausdruck

$$\begin{aligned} \log(p(O^k|\Lambda)) &= \log\left(\sum_{i=1}^{C} p(O^k, i|\Lambda)\right) \\ &= \log\left(\sum_{i=1}^{C} p(O^k|i,\Lambda)\, pr(i|\Lambda)\right) \end{aligned}$$

mit der Reestimierung steigt und somit die Zielfunktion fällt.

Ein alternativer Ansatz zur Lösung von Problem 5 wird im Abschnitt 3.2 vorgestellt. Dort wird ein Mischmodell-Ansatz vorgenommen, bei dem beim Training auf die Bildung von Sequenzpartitionen gänzlich verzichtet werden kann.

Ein generelles Problem bei den meisten Clusteralgorithmen stellt die Wahl einer geeigneten Clusteranzahl dar. In diesem Abschnitt wurde die Problematik dadurch ausgeklammert, dass die Anzahl vorgegeben wurde. Lösungsansätze zur Bestimmung einer geeigneten Clusteranzahl werden im Abschnitt 5.4 vorgestellt.

2.4 Erweiterungen

Übergangsklassen
Eine wesentliche Erweiterung von Hidden-Markov-Modellen, die in dieser Arbeit zur Modellierung von Bausparkollektiven verwendet wird, besteht in der Einführung so genannter diskreter Übergangsklassen. Hinter diesem Konzept, das ausführlich in [23] beschrieben wird, verbirgt sich Folgendes: Bei Vorgabe von L Übergangsklassen wird die Übergangsmatrix A der Standardmodellierung erweitert auf einen Vektor von Übergangsmatrizen:

$$A \quad \rightarrow \quad (A_1, \ldots, A_L).$$

Wird das Hidden-Markov-Modell als Graph betrachtet, so bedeutet dies, dass jede Übergangskante zwischen zwei Zuständen durch L parallele Kanten ersetzt wird.

Zu jedem Zeitpunkt befindet sich das System wie bisher in genau einem Zustand und jetzt zusätzlich in genau einer Übergangsklasse. Die Wahrscheinlichkeiten der Nachfolgezustände hängen vom aktuellen Zustand und von der aktuellen Übergangsklasse ab. Realisiert wird dies durch eine eigene Übergangsmatrix für jede Übergangsklasse. Die Wahl der jeweiligen Übergangsklasse zu einem Zeitpunkt wird deterministisch von der bis zu diesem Zeitpunkt reichenden Anfangsteilsequenz bestimmt. Es wird also eine Funktion eingeführt, die für jedes t mit $1 \leq t \leq T$ eine Anfangssequenz

$$O_{(t)} := (O_1, \ldots, O_t)$$

auf die Menge der Übergangsklassen abbildet:

$$\text{class} : \{O_{(t)}\} \rightarrow \{1, \ldots, L\}. \tag{2.39}$$

Diese Erweiterung hat Auswirkungen auf fast alle HMM-Algorithmen, da sie die Markov-Eigenschaft (2.2) verändert:
Ohne Übergangsklassen:

$$pr(q_j|q_i) = a_{ij}.$$

Mit Übergangsklassen:

$$pr(q_j|q_i, \text{class}(O_{(t)}) = l) = (a_{ij})_l,$$

wobei $(a_{ij})_l$ das Element (i,j) der l-ten Übergangsmatrix kennzeichnet. In [23] wird gezeigt, wie die Parameterreestimierung durch den Baum-Welch-Algorithmus beim bisherigen Modell erweitert werden muss, sodass sie auch für dieses erweiterte Modell Gültigkeit hat.

Der Grund, diese Erweiterung durch Übergangsklassen bei der HMM-Modellierung von Bausparzeitreihen einzusetzen liegt darin, dass das Verhalten der Bausparer zu einem gegebenen Zeitpunkt gewissen deterministischen Randbedingungen unterworfen ist, die sich wiederum aus den bisherigen Aktionen des Bausparers ergeben. Mit der Einführung der Übergangsklassen wird dem Modell, salopp gesprochen, ein „Gedächtnis“ zur Verfügung gestellt, das bisher ausgegebene Symbole in abstrakter Form speichern kann. Dieses „Gedächtnis“ führt dazu, dass die Ausgabe von zeitlich weiter zurückliegenden Symbolen Einfluss auf die Wahrscheinlichkeitsverteilung des aktuellen Ausgabesymbols hat. Besondere Bedeutung gewinnt dieser Umstand bei der Generierung künstlicher Sequenzen. Erneut wird hierauf im Abschnitt 4.5 eingegangen.

Abgeschnittene Normalverteilung
Neben der bereits erwähnten Gauß-Dichte wird in dieser Arbeit auch eine asymmetrische, im negativen Wertebereich auf null gesetzte Gauß-Dichte verwendet. Mit dieser gestutzten Dichte ist sichergestellt, dass in den generierten Sequenzen ausschließlich Werte vorkommen, die größer oder gleich null sind. In [23] wird ausführlich gezeigt, wie die Baum-Welch-Reestimierungsformeln für diese Form der Ausgabefunktionen modifiziert werden müssen.

Ausgabefixierung
Im Kapitel 4 zur konkreten Modellbildung wird deutlich werden, dass es sinnvoll ist, besondere Zustände einzuführen, die sich dadurch von den anderen Zuständen unterscheiden, dass sie nur ein ganz bestimmtes Symbol ausgeben können. Für diese Zustände wurde der Trainingsalgorithmus dahingehend erweitert, dass die Ausgabeparameter fixiert werden können, also vom Training nicht veränderbar sind. Diese Zustände werden beim Training bereits so initialisiert, dass sie nur das geforderte Symbol ausgeben können. Dies kann auch bei einem kontinuierlichen Modell mit ausreichender Genauigkeit erreicht werden, wenn die Varianz auf einen sehr kleinen Wert gesetzt wird, sodass die Wahrscheinlichkeit einer Ausgabe von Symbolen, die stark vom Mittelwert abweichen, verschwindend gering ist.

2.5 EM-Algorithmus

In diesem Abschnitt soll der „Expectation Maximization“-Algorithmus (kurz EM-Algorithmus) vorgestellt werden, der im Kapitel 3 zur Parameterschätzung in Mischmodellen verwendet wird. Hierbei handelt es sich um ein sehr allgemeines Schema zur Maximum-Likelihood-Schätzung, das in [8] erstmals vorgestellt wurde. Eine ausführliche Darstellung des EM-Algorithmus liegt mit [30] vor, einem Buch, das sich ausschließlich diesem Thema widmet.

Zunächst wird der Algorithmus in allgemeiner Form hergeleitet. Anschließend wird bewiesen, dass der Algorithmus gegen einen festen Wert der Likelihood-Funktion konvergiert, der unter weiteren, wenig einschränkenden Annahmen einem Maximum der Likelihood-Funktion entspricht. Im letzten Teil wird gezeigt, welche Form der EM-Algorithmus für den Fall der Parameterschätzung bei Hidden-Markov-Modellen annimmt und dass sich aus seiner Anwendung

genau die bekannten Reestimierungsformeln des Baum-Welch-Algorithmus aus Abschnitt 2.3.3 ergeben.

Annahmen und Problemstellung
Gegeben sei eine Stichprobe X eines kontinuierlichen Zufallsvektors x vom Umfang K:

$$X = (x^1, \ldots, x^K).$$

Es wird angenommen, dass sich jeder Vektor x^k (im Folgenden auch vollständiges Datum genannt) zusammensetzt aus einer beobachtbaren Komponente y^k und einer versteckten Komponente z^k

$$x^k = (y^k, z^k), \quad y^k \text{ beobachtbar}, \quad z^k \text{ versteckt}$$

und dass die Vektoren gemäß der folgenden Dichtefunktionen verteilt sind:[1]

$$p(x|\Theta), \quad p(y|\Theta), \quad p(z|\Theta).$$

Folgende abkürzende Notationen werden im weiteren Verlauf verwendet:

$$Y = (y^1, \ldots, y^K), \quad Z = (z^1, \ldots, z^K), \quad X = (Y, Z).$$

Ziel des EM-Algorithmus ist die Maximierung der Log-Likelihoodfunktion $\log L_Y(\Theta)$ der beobachtbaren Daten, die sich wie folgt berechnen läßt:

$$\log L_Y(\Theta) := \log p(Y|\Theta) = \sum_{k=1}^{K} \log(p(y^k|\Theta)). \tag{2.40}$$

In der Regel wird es nicht möglich sein, die Gleichung (2.40) analytisch zu maximieren, sodass der Einsatz eines numerischen Verfahrens notwendig ist. Beim nachfolgend vorgestellten EM-Algorithmus handelt es sich um ein iteratives Verfahren, dass die Maximierung von (2.40) indirekt über die Maximierung einer Hilfsfunktion (Q-Funktion) bewerkstelligt.

Bei der Berechnung der Q-Funktion spielt die Gesamt-Log-Likelihoodfunktion $\log L_X(\Theta)$ der vollständigen Daten eine wichtige Rolle, die wie folgt angegeben werden kann:

$$\begin{aligned} \log L_X(\Theta) &:= \log p(X|\Theta) \\ &= \sum_{k=1}^{K} \log(p(x^k|\Theta)) \\ &= \sum_{k=1}^{K} \log(p(y^k, z^k|\Theta)). \end{aligned} \tag{2.41}$$

Dieser Ausdruck ist jedoch wegen der Unkenntnis der z^k nicht berechenbar, lediglich Erwartungswerte können bestimmt werden.

[1] Obwohl es sich um verschiedene Funktionen handelt, sollte die gleiche Bezeichnung nicht zu Konfusionen führen, da durch die Angabe der Variablen klar wird, was gemeint ist.

Q-Funktion
Sei Θ' ein vorgegebener Parameter von $p(x|\Theta)$. Unter Verwendung dieses Parameters ist die Q-Funktion wie folgt definiert:

$$Q(\Theta, \Theta') := E_z \left[\log \left(L_X(\Theta) \right) | \Theta', Y \right] . \tag{2.42}$$

Dieser Ausdruck bedarf einiger Erklärung und soll daher jetzt genauer betrachtet werden. Zunächst ist zu beachten, dass $Q(\Theta, \Theta')$ eine Funktion von Θ ist, dem Paramter, der die Verteilung der vollständigen Daten festlegt, während Θ' einen vorgegebenen konstanten Parametervektor bezeichnet. Die Q-Funktion ist der Erwartungswert der Gesamt-Log-Likelihood, der bezüglich der bedingten Verteilung von z bei gegebenem Y und bei Vorliegen des Parameters Θ' berechnet wird:

$$\begin{aligned} Q(\Theta, \Theta') &= E_z \left[\sum_{k=1}^{K} \log(p(y^k, z|\Theta)) \,\middle|\, \Theta', Y \right] \\ &= \sum_{k=1}^{K} E_z \left[\log(p(z|\Theta, y^k)) | \Theta', y^k \right] \\ &= \sum_{k=1}^{K} \int \log(p(z|\Theta, y^k))\, p(z|\Theta', y^k)\, dz. \end{aligned}$$

Mittels dieser Q-Funktion kann der EM-Algorithmus folgendermaßen in seiner allgemein gültigen Form angegeben werden:

EM-Algorithmus

Eingabe: Daten $Y = (y^1, \ldots, y^K)$, Dichtefunktionen $p(x|\Theta), p(y|\Theta), p(z|\Theta)$
Initialparameter Θ^0, max. Iterationszahl I, Konvergenzschranke EPS

$Z_{old} := \log(L_Y(\Theta^0))$
for $i := 0$ **to** I **do**
1. **E-Schritt**

$$Q(\Theta, \Theta^i) := E_z \left[\log(L_X(\Theta)) | \Theta^i, Y \right] \tag{2.43}$$

2. **M-Schritt**

$$\Theta^{i+1} := \arg\max_{\Theta} Q(\Theta, \Theta^i) \tag{2.44}$$

3. $Z_{new} := \log(L_Y(\Theta^{i+1}))$
4. **if** $(Z_{new} - Z_{old} > \text{EPS})$ **then**
   $Z_{old} := Z_{new}$
   else

Abbruch
end if
end for
Ausgabe: optimaler Parametervektor Θ

Der Algorithmus besteht also in der abwechselnden Berechnung der Q-Funktion (E-Schritt) und deren Maximierung (M-Schritt). Auf diese Art und Weise wird eine Folge von Parametervektoren Θ^i definiert, die abgebrochen wird, wenn die Differenz der Log-Likelihood zweier aufeinanderfolgender Θ-Werte nach Gleichung (2.40) unterhalb einer vorgegebenen Schranke EPS gefallen ist oder wenn die maximale Iterationszahl erreicht ist.

Konvergenz
Der folgenden Satz stellt sicher, dass die Likelihood durch den EM-Algorithmus nicht verschlechtert wird. Der anschließende Beweis folgt dem Vorgehen in [30].

Satz 2.5.1 *Seien Θ^i und Θ^{i+1} zwei Parametervektoren aus der Folge, die durch den EM-Algorithmus definiert wird, und $L_Y(\cdot)$ die Likelihood definiert durch Gleichung* (2.40)*. Dann gilt:*

$$\log(L_Y(\Theta^{i+1})) \geq \log(L_Y(\Theta^i)). \tag{2.45}$$

Beweis:
Für einen beliebigen Parameter Θ gilt:

$$L_Y(\Theta) = p(Y|\Theta) = \frac{p(Y,Z|\Theta)}{p(Z|\Theta,Y)} = \frac{L_X(\Theta)}{p(Z|\Theta,Y)}.$$

Hieraus folgt:

$$\log(L_Y(\Theta)) = \log(L_X(\Theta)) - \log(p(Z|\Theta,Y)).$$

Von dieser Gleichung wird auf beiden Seiten der bedingte Erwartungswert bezüglich z bei gegebenem Y unter Verwendung des Parameters Θ^i gebildet. Die linke Seite dieser Gleichung ist von z unabhängig, es gilt also:

$$E_z\left[\log(L_Y(\Theta))|Y,\Theta^i\right] = \log(L_Y(\Theta)).$$

Unter Verwendung der Definition der Q-Funktion (2.42) ergibt sich somit:

$$\begin{aligned}\log(L_Y(\Theta)) &= Q(\Theta,\Theta^i) - E_z\left[\log(p(Z|\Theta,Y))|Y,\Theta^i\right] \\ &=: Q(\Theta,\Theta^i) - H(\Theta,\Theta^i).\end{aligned} \tag{2.46}$$

Dies wiederum bedeutet, dass

$$\begin{aligned}&\log(L_Y(\Theta^{i+1})) - \log(L_Y(\Theta^i)) \\ &= \left(Q(\Theta^{i+1},\Theta^i) - Q(\Theta^i,\Theta^i)\right) - \left(H(\Theta^{i+1},\Theta^i) - H(\Theta^i,\Theta^i)\right).\end{aligned} \tag{2.47}$$

Zum Beweis des Satzes ist zu zeigen, dass dies größer oder gleich null ist. Aus dem M-Schritt im Algorithmus folgt direkt, dass

$$Q(\Theta^{i+1}, \Theta^i) - Q(\Theta^i, \Theta^i) \geq 0. \tag{2.48}$$

Somit bleibt zu zeigen, dass der zweite Term in Gleichung (2.47) kleiner oder gleich null ist. Für jedes Θ und somit auch für Θ^{i+1} gilt:

$$\begin{aligned}
& H(\Theta, \Theta^i) - H(\Theta^i, \Theta^i) \\
&= E_z\left[\log(p(Z|\Theta, Y)) \mid Y, \Theta^i\right] - E_z\left[\log(p(Z|\Theta^i, Y)) \mid Y, \Theta^i\right] \\
&= E_z\left[\log\left(p(Z|\Theta, Y)/p(Z|\Theta^i, Y)\right) \mid Y, \Theta^i\right] \\
&\leq \log\left(E_z\left[p(Z|\Theta, Y)/p(Z|\Theta^i, Y) \mid Y, \Theta^i\right]\right) \qquad (2.49) \\
&= \log \int_{z\in\mathcal{Z}} \frac{p(z|\Theta, Y)}{p(z|\Theta^i, Y)}\, p(z|\Theta^i, Y)\, dz \\
&= \log \int_{z\in\mathcal{Z}} p(z|\Theta, Y)\, dz \\
&= 0. \qquad (2.50)
\end{aligned}$$

□

Die Ungleichung (2.49) ergibt sich aus der Konkavität des Logarithmus und aus der Jensen Ungleichung [14]. Somit ist also gezeigt, dass der EM-Algorithmus die Likelihood der beobachteten Daten Y in keinem Schritt verschlechtern kann. Wenn die Likelihood nach oben beschränkt ist und wenn für den Initialparameter Θ^0 gilt:

$$\log(L_Y(\Theta^0)) > -\infty, \tag{2.51}$$

dann konvergiert der EM-Algorithmus, und die durch ihn definierte Folge von Log-Likelihood-Werten strebt gegen einen festen Wert $L^\star$. In [42] wird gezeigt, dass dieser feste Wert unter Voraussetzung weiterer, recht schwacher Annahmen auch ein lokales oder globales Maximum der Likelihood-Funktion ist.

Bemerkung 2.5.2 *Die Aussage von Satz 2.5.1 gilt auch, wenn im M-Schritt des EM-Algorithmus, gegeben durch Gleichung* (2.44)*, sichergestellt ist, dass die Q-Funktion nicht verschlechtert wird, wenn also gilt:*

$$Q(\Theta^{i+1}, \Theta^i) \geq Q(\Theta^i, \Theta^i).$$

In diesem Fall spricht man vom generalisierten EM-Algorithmus.

Beweis
Da die Ungleichung (2.48) auch in dieser generalisierten Version gilt, kann der Beweis des EM-Algorithmus auch zum Beweis der Bemerkung verwendet werden. □

Diese Bemerkung kann in der praktischen Anwendung große Bedeutung haben, wenn die Maximierung der Q-Funktion nicht oder nur mit großem Aufwand möglich ist, während ihre Vergrößerung einfach sicherzustellen ist.

Anwendung auf Hidden-Markov-Modelle

Nach dem Konvergenzbeweis soll jetzt gezeigt werden, wie der Algorithmus auf die ML-Schätzung bei Hidden-Markov-Modellen angewendet werden kann. Die Darstellung folgt weitgehend der Herleitung in [5]. Es wird sich zeigen, dass die Formeln, die sich daraus ergeben, mit den Reestimierungsformeln des Baum-Welch-Algorithmus übereinstimmen.

Die sichtbaren (unvollständigen) Daten sind durch eine Menge von Sequenzen gegeben:

$$O = (O^1, \ldots, O^K)$$

mit den Längen $(T^1, \ldots, T^K)$.

Die damit korrespondierenden Zustandspfade

$$Q = (Q^1, \ldots, Q^K)$$

mit den jeweils gleichen Längen übernehmen die Rolle der versteckten Daten [2] und die damit korrespondierende Variable wird mit q gekennzeichnet. Die vollständige Likelihood-Funktion hat, wieder unter der Annahme, dass die Sequenzen voneinander unabhängig sind, folgende Gestalt:

$$L_C(\lambda) := \prod_{k=1}^{K} p(Q^k, O^k|\lambda).$$

Daraus ergibt sich folgende Q-Funktion:

$$\begin{aligned} Q(\lambda, \lambda^i) &= E_q\left[\log(L_C(\lambda)) \mid O, \lambda^i\right] \\ &= E_q\left[\sum_{k=1}^{K} \log(p(Q^k, O^k|\lambda)) \,\middle|\, O, \lambda^i\right] \\ &= \sum_{k=1}^{K} E_q\left[\log(p(Q^k, O^k|\lambda)) \,\middle|\, O^k, \lambda^i\right] \\ &= \sum_{k=1}^{K} \sum_{q \in \mathcal{Q}_{T^k}} \log(p(q, O^k|\lambda)) pr(q|O^k, \lambda^i). \end{aligned} \tag{2.52}$$

Die innere Summe in (2.52) erstreckt sich über $\mathcal{Q}_{T^k}$, womit die Menge aller möglichen Zustandspfade q der Länge T^k bezeichnet wird. In der letzten Gleichung wird schließlich noch $pr(q|O^k, \lambda^i)$ durch $p(q, O^k|\lambda^i)$ ersetzt. Dies ist bezüglich der Maximierung der Q-Funktion äquivalent, da gilt:

$$pr(q|O^k, \lambda^i) = \frac{p(q, O^k|\lambda^i)}{p(O^k|\lambda^i)}$$

[2] Man beachte die unterschiedliche Bedeutung von Q zur Bezeichnung der Q-Funktion und der Menge der Zustandspfade.

und $p(O^k|\lambda^i)$ unabhängig von λ ist. Daraus ergibt sich folgende modifizierte Q-Funktion:

$$\begin{aligned}\tilde{Q}(\lambda, \lambda^i) &= \sum_{k=1}^{K} \sum_{q \in \mathcal{Q}_{T^k}} \log(p(q, O^k|\lambda)) p(q, O^k|\lambda^i) \\ &= \sum_{k=1}^{K} \sum_{q \in \mathcal{Q}_{T^k}} \log \left(\pi_{q_1} f_{q_1}(O_1^k) \prod_{t=2}^{T^k} \left(a_{q_{t-1}q_t} f_{q_t}(O_t^k) \right) \right) p(q, O^k|\lambda^i).\end{aligned}$$

Somit läßt sich die Q-Funktion in drei getrennt zu maximierende Terme aufspalten:

$$\begin{aligned}\tilde{Q}(\lambda, \lambda^i) &= \sum_{k=1}^{K} \sum_{q \in \mathcal{Q}_{T^k}} \log(\pi_{q_1}) p(q, O^k|\lambda^i) \\ &+ \sum_{k=1}^{K} \sum_{q \in \mathcal{Q}_{T^k}} \sum_{t=2}^{T^k} \log(a_{q_{t-1}q_t}) p(q, O^k|\lambda^i) \\ &+ \sum_{k=1}^{K} \sum_{q \in \mathcal{Q}_{T^k}} \sum_{t=1}^{T^k} \log(f_{q_t}(O_t^k)) p(q, O^k|\lambda^i). \end{aligned} \tag{2.53}$$

Beispielhaft sei jetzt der erste Ausdruck für die Initialwahrscheinlichkeit genauer untersucht:

$$\begin{aligned}&\sum_{k=1}^{K} \sum_{q \in \mathcal{Q}_{T^k}} \log(\pi_{q_1}) p(q, O^k|\lambda^i) \\ &= \sum_{k=1}^{K} \sum_{q_1=1}^{N} \cdots \sum_{q_{T^k}=1}^{N} \log(\pi_{q_1}) p(q_1|O^k, q_2, \ldots, q_{T^k}, \lambda^i)\, p(O^k, q_2, \ldots, q_{T^k}|\lambda^i) \\ &= \sum_{k=1}^{K} \sum_{q_1=1}^{N} \cdots \sum_{q_{T^k}=1}^{N} \log(\pi_{q_1}) p(q_1|O^k, \lambda^i)\, p(O^k, q_2, \ldots, q_{T^k|\lambda^i}) \\ &= \sum_{k=1}^{K} \sum_{q_1=1}^{N} \log(\pi_{q_1}) p(q_1|O^k, \lambda^i) \sum_{q_2=1}^{N} \cdots \sum_{q_{T^k}=1}^{N} p(O^k, q_2, \ldots, q_{T^k}|\lambda^i) \\ &= \sum_{k=1}^{K} \sum_{q_1=1}^{N} \log(\pi_{q_1}) p(q_1|O^k, \lambda^i)\, p(O^k|\lambda^i) \\ &= \sum_{k=1}^{K} \sum_{q_1=1}^{N} \log(\pi_{q_1}) p(q_1, O^k|\lambda^i) \\ &= \sum_{q_1=1}^{N} \log(\pi_{q_1}) \sum_{k=1}^{K} p(q_1, O^k|\lambda^i). \end{aligned} \tag{2.54}$$

Für den M-Schritt des Algorithmus muss der letzte Ausdruck maximiert werden unter der Ne-

benbedingung:

$$\sum_{j=1}^{N} \pi_j = 1. \tag{2.55}$$

Dies geschieht mit der Methode der Lagrange-Multiplikatoren und führt für $l = 1, \ldots, N$ zu folgender Gleichung:

$$\frac{\partial}{\partial \pi_l} \left[\sum_{j=1}^{N} \left(\log(\pi_j) \sum_{k=1}^{K} p(q_1 = j, O^k | \lambda^i) \right) + \gamma \left(\sum_{j=1}^{N} \pi_j - 1 \right) \right] = 0. \tag{2.56}$$

Die Ableitung berechnet ergibt folgenden Ausdruck für π_l:

$$\pi_l = -\frac{1}{\gamma} \sum_{k=1}^{K} p(q_1 = l, O^k | \lambda^i). \tag{2.57}$$

Wird diese Gleichung über alle Zustände summiert, so ergibt sich mit Einsetzen der Nebenbedingung (2.55) für den Lagrange-Parameter:

$$\gamma = -\sum_{j=1}^{N} \sum_{k=1}^{K} p(q_1 = j, O^k | \lambda^i). \tag{2.58}$$

Das so errechnete γ kann in Gleichung (2.57) eingesetzt werden und ergibt schließlich den maximalen Wert für π_l:

$$\begin{aligned} \pi_l &= \frac{\sum_{k=1}^{K} p(q_1 = l, O^k | \lambda^i)}{\sum_{k=1}^{K} \sum_{j=1}^{N} p(q_1 = j, O^k | \lambda^i)} \\ &= \frac{\sum_{k=1}^{K} \alpha_1^k(l) \beta_1^k(l)}{\sum_{k=1}^{K} \sum_{j=1}^{N} \alpha_1^k(j) \beta_1^k(j)}. \end{aligned} \tag{2.59}$$

Dies ist aber genau die Reestimierungsformel (2.28) des Baum-Welch-Algorithmus für π_l. Die übrigen Reestimierungsformeln ergeben sich ganz analog ausgehend von der Q-Funktion nach Gleichung (2.53). Auf ihre Herleitung wird hier verzichtet, sie findet sich in [5]. Wird mit einer Mischung von Dichtefunktionen in der Ausgabe gearbeitet, so müssen die versteckten Daten und damit auch die Q-Funktion noch entsprechend erweitert werden. Das Vorgehen ist dann aber analog zu dem gezeigten, und auch in diesem Fall ergeben sich die bekannten Reestimierungsformeln.

Somit wurden in diesem Kapitel die wesentlichen Elemente kontinuierlicher Hidden-Markov-Modelle und die wichtigsten Algorithmen zur Lösung grundlegender HMM-Probleme vorgestellt. Die recht ausführliche Darstellung des EM-Algorithmus stellte einerseits die Baum-Welch-Reestimierungsformeln auf ein solides Fundament und ist andererseits auch die Basis des Parameter-Trainings in Mischmodellen, welches im nachfolgenden Kapitel beschrieben wird.

Kapitel 3

Mischmodelle zur Clusterung von Sequenzen und zusammengesetzten Daten

Die Darstellung heterogener Daten durch so genannte endliche Mischmodelle spielt in vielen Anwendungen eine besonders wichtige Rolle. Es handelt sich um einen Modellansatz, der in die Kategorie „modellbasiertes Clustern“ fällt. Das bedeutet, dass die Clusterung der Daten, also die Trennung in Subpopulationen unter der Prämisse einer bestimmten Modellfamilie durchgeführt wird. Ein Grund für die weite Verbreitung der Mischmodelle ist u. a. die Tatsache, dass mit dem im Abschnitt 2.5 dargestellten EM-Algorithmus ein effektives Verfahren zur ML-Schätzung in dieser Modellklasse zur Verfügung steht. Eine gute und umfassende Übersicht dieser Modelle findet sich z. B. in [29].

Im ersten Abschnitt dieses Kapitels wird die Mischmodellierung in allgemeiner Form vorgestellt. Es wird gezeigt, wie die ML-Schätzung in dieser Modellklasse mit Hilfe des EM-Algorithmus durchgeführt werden kann. Daran anschliessend wird demonstriert, wie sich Mischmodelle mit Hidden-Markov-Modellen kombinieren lassen und so zur Clusterung von Sequenzen eingesetzt werden können. Es wird der Algorithmus „MIX-HMM“ präsentiert und dessen Konvergenz bewiesen. Dieser Algorithmus stellt eine Alternative zum HMM-Clusteralgorithmus aus Abschnitt 2.3.5 dar. Im dritten Teil dieses Kapitels wird der MIX-HMM-Algorithmus dahingehend erweitert, dass Sequenzen in Kombination mit weiteren, zeitunabhängigen Merkmalen geclustert werden können. Hierdurch ist es z. B. möglich, in der Clusterung von Bauspardaten die Bausparsumme eines Vertrages als Merkmal mit zu berücksichtigen. Das Kapitel schließt mit einigen Bemerkungen über die Rolle von Abstandsmaßen und Wahrscheinlichkeitsdichten beim Clustern.

3.1 Mischmodelle

Modellansatz
In diesem Abschnitt werden zunächst Mischmodelle in allgemeiner Form und das Problem der

ML-Schätzung in Mischmodellen dargestellt. Im Anschluss daran wird gezeigt, wie sich das Problem mit dem EM-Algorithmus lösen lässt.

Gegeben ist eine Menge von K unabhängigen, kontinuierlichen Datenvektoren:

$$Y = (y^1, \ldots, y^K),$$

wobei die Dimension der Datenvektoren in diesem Zusammenhang unerheblich ist. Ferner sind M Wahrscheinlichkeitsdichten mit Parametervektoren Θ_j

$$p_j(y|\Theta_j), \qquad 1 \leq j \leq M$$

gegeben, aus denen durch Überlagerung das so genannte Mischmodell entsteht:

$$p(y|\Theta) = \sum_{j=1}^{M} \alpha_j p_j(y|\Theta_j), \tag{3.1}$$

mit dem Gesamtparametervektor

$$\Theta = (\alpha_1, \ldots, \alpha_M, \Theta_1, \ldots, \Theta_M).$$

Die Gewichte der Komponentendichten α_j müssen folgende Bedingungen erfüllen:

$$\sum_{j=1}^{M} \alpha_j = 1, \quad \alpha_j \geq 0 \qquad \text{f. a. } j.$$

Abbildung 3.1 veranschaulicht diesen Modellansatz für den einfachen Fall zweidimensionaler Daten und drei Gaußdichten als Komponentendichten. Die konzentrischen Linien sind Höhenlinien der einzelnen Wahrscheinlichkeitsdichten und die Dichte des eingezeichneten Punktes y setzt sich zusammen aus Anteilen aller drei Komponentendichten. Somit findet im Gegensatz zu geometrischen Clusterverfahren keine strikte Aufteilung des Raumes auf die verschiedenen Cluster statt.

Der Modellierungsansatz in Form von Gleichung (3.1) kann auf zwei verschiedene Arten interpretiert werden:

1. Es wird angenommen, dass die gegebenen Daten Y der Gesamtdichte $p(y|\Theta)$ entstammen, und die Aufgabe besteht darin, den optimalen Parameter Θ bei gegebenen Daten Y zu schätzen (ML-Schätzung).
2. Es wird angenommen, dass die Daten aus M Untergesamtheiten entstammen, die durch die M verschiedenen Komponentendichten $p_j(y|\Theta_j)$ repräsentiert werden, wobei die Zugehörigkeit eines Datums zu einer Subpopulation unbekannt ist. Aufgabe ist es, die Daten mit dem gegebenen Modellansatz optimal zu clustern. Da die Aufgabe über das bloße Partitionieren der Daten hinausgeht, wird die Lösung dieser Aufgabe auch als modellbasiertes Clustern bezeichnet.

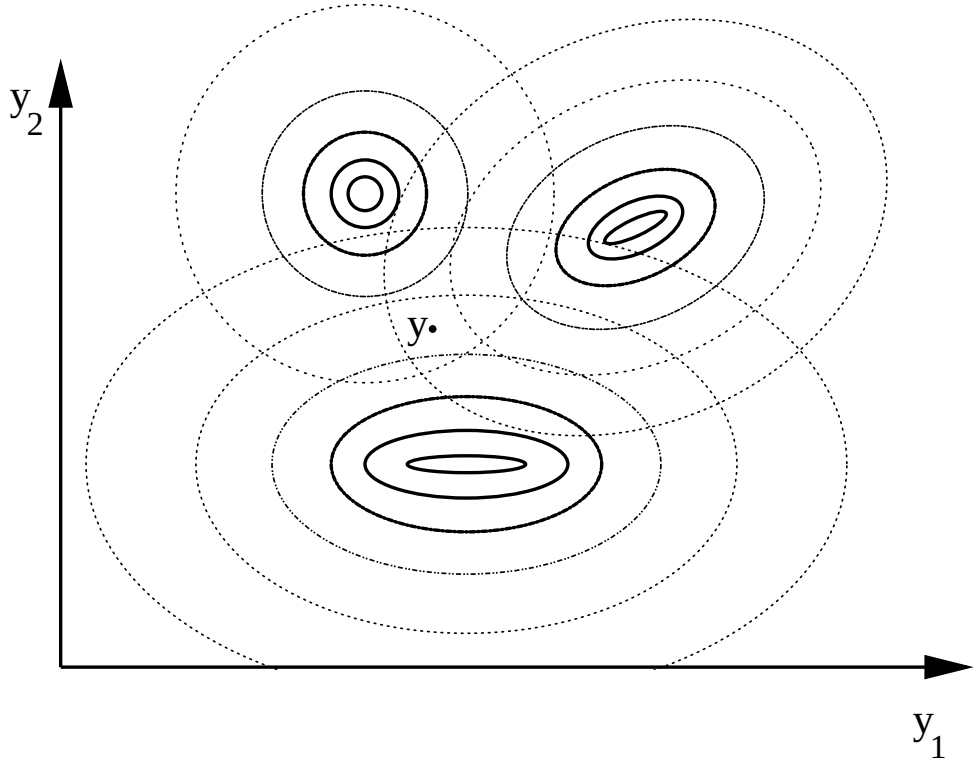

Abbildung 3.1: *Höhenlinien der Wahrscheinlichkeitsdichte in einem einfachen Mischmodell mit drei Komponentendichten ($M = 3$) für zweidimensionale Daten $y = (y_1, y_2)$.*

Da wie eingangs erwähnt die Modellierung mit Mischmodellen als Alternative zum HMM-Clustern aus Abschnitt 2.3.5 untersucht werden soll, wird in dieser Arbeit die zweite Interpretation im Vordergrund stehen. Dies sind aber lediglich zwei unterschiedliche Betrachtungsweisen, in beiden Fällen gilt es, folgendes Problem zu lösen:

Problem 6 *Maximiere bei gegebenen Daten $Y = (y^1, \ldots, y^K)$ die folgende Log-Likelihood-Funktion:*

$$\begin{aligned} \log(L_Y(\Theta)) &= \log(p(Y|\Theta)) \\ &= \log\left(\prod_{k=1}^{K} p(y^k|\Theta)\right) \\ &= \sum_{k=1}^{K} \log\left(\sum_{j=1}^{M} \alpha_j p_j(y^k|\Theta_j)\right). \end{aligned} \tag{3.2}$$

Das bedeutet, bestimme $\hat{\Theta}$ so, dass Folgendes gilt:

$$\hat{\Theta} = \arg\max_{\Theta} \log(L_Y(\Theta)). \tag{3.3}$$

Lösung mit dem EM-Algorithmus

Für das Maximierungsproblem (3.3) ist keine analytische Lösung bekannt. Es ist aber möglich, iterativ eine zumindest lokal optimale Lösung mit Hilfe des in Abschnitt 2.5 definierten EM-Algorithmus zu finden. Zu diesem Zweck wird das Problem vereinfacht. Es wird angenommen, dass jedes Datum y^k mit einem zusätzlichen Merkmal z^k verknüpft ist, das die Clusterzugehörigkeit angibt. Also bedeutet $z^k = j$, dass das k-te Objekt vom j-ten Cluster stammt, bzw. vom j-ten Komponentenmodell erzeugt wurde. Das zusätzliche Merkmal z^k stellt in dieser Anwendung das versteckte Datum dar. Mit $Z := (z^1, \ldots, z^K)$ nimmt die vollständige Log-Likelihood bei gegebenem Y und Z folgende Gestalt an:

$$\log(L_C(\Theta)) := \log(p(Y, Z|\Theta)) = \sum_{k=1}^{K} \log(\alpha_{z^k} p_{z^k}(y^k|\Theta_{z^k})). \tag{3.4}$$

Von der Summe im Logarithmus aus Gleichung (3.2) bleibt also für jedes Objekt nur der Term mit der „wahren“ Wahrscheinlichkeitsdichte bestehen. Mit Gleichung (2.43) und Gleichung (3.4) ergibt sich für die Q-Funktion der folgende Ausdruck:

$$\begin{aligned}
Q(\Theta, \Theta^i) &= E_Z\left[\log(p(Y, Z|\Theta) \mid \Theta^i, Y)\right] \\
&= \sum_{k=1}^{K} E_Z\left[\log(\alpha_{z^k} p_{z^k}(y^k|\Theta_{z^k})) \,|\Theta^i, y^k\right] \\
&= \sum_{k=1}^{K}\sum_{j=1}^{M} \log(\alpha_j p_j(y^k|\Theta_j))\, p(z^k = j|\Theta^i, y^k) \\
&= \sum_{k=1}^{K}\sum_{j=1}^{M} \log(\alpha_j)\, p(z^k = j|\Theta^i, y^k) + \\
&\quad \sum_{k=1}^{K}\sum_{j=1}^{M} \log(p_j(y^k|\Theta_j))\, p(z^k = j|\Theta^i, y^k).
\end{aligned} \tag{3.5}$$

Der Ausdruck (3.5) muss zur Durchführung des M-Schrittes bezüglich Θ maximiert werden, während Θ^i als konstant angesehen wird. Aus der letzten Gleichung ist ersichtlich, dass sich die Q-Funktion in zwei voneinander unabhängige Terme aufspalten lässt, deren Maximierung getrennt durchgeführt werden kann.

Die Maximierung von

$$\begin{aligned}
&\sum_{j=1}^{M}\sum_{k=1}^{K} \log(\alpha_j) p(z^k = j|\Theta^i, y^k) \\
&= \sum_{j=1}^{M} \log(\alpha_j) \sum_{k=1}^{K} p(z^k = j|\Theta^i, y^k)
\end{aligned} \tag{3.6}$$

bezüglich der Komponentengewichte α_j unter der Nebenbedingung $\sum_{j=1}^{M} \alpha_j = 1$ wurde bereits im Abschnitt 2.5 hergeleitet, denn die Maximierung von Gleichung (2.54) besitzt genau die gleiche Struktur wie die hier gestellte Aufgabe. Somit kann auch der Ausdruck in Gleichung (3.6) mit der Methode der Lagrange-Parameter maximiert werden und die Lösung lautet

$$\begin{aligned} \alpha_j &= \frac{\sum_{k=1}^{K} p(z^k = j|\Theta^i, y^k)}{\sum_{k=1}^{K}\sum_{l=1}^{M} p(z^k = l|\Theta^i, y^k)} \\ &= \frac{1}{K}\sum_{k=1}^{K} p(z^k = j|\Theta^i, y^k). \end{aligned} \tag{3.7}$$

Diese Gleichung besitzt eine ganz anschauliche Interpretation: Die Gewichtung der j-ten Mischkomponente ist gleich der mittleren Wahrscheinlichkeit der j-ten Mischkomponente, gemittelt über alle Daten, bei gegebenen Parametern Θ^i. Die Wahrscheinlichkeit einer Mischkomponente bei gegebenem Datum y^k kann mit Hilfe der Bayes-Formel berechnet werden und mit der abkürzenden Schreibenweise

$$p(y^k|\Theta^i, z^k = j) =: p_j(y^k|\Theta_j^i)$$

ergibt sich hierfür

$$\begin{aligned} p(z^k = j|\Theta^i, y^k) &= \frac{p(z^k = j|\Theta^i)p(y^k|\Theta^i, z^k = j)}{p(y^k|\Theta^i)} \\ &= \frac{\alpha_j^i p_j(y^k|\Theta_j^i)}{\sum_{l=1}^{M} \alpha_l^i p_l(y^k|\Theta_l^i)}. \end{aligned} \tag{3.8}$$

Somit verbleibt zur Durchführung des M-Schrittes noch die Maximierung von

$$\sum_{k=1}^{K}\sum_{j=1}^{M} \log(p_j(y^k|\Theta_j))\, p(z^k = j|\Theta^i, y^k) \tag{3.9}$$

aus Gleichung (3.5), was natürlich von der expliziten Form der angenommenen Komponentendichte $p_j(y|\Theta_j)$ abhängt.

Normalverteilung

In vielen Anwendungen wird eine multivariate Normalverteilung $\mathcal{N}(y, \mu_j, \Sigma_j)$, gegeben durch Gleichung (2.24), angenommen. In diesem Fall können geschlossene Lösungformeln angegeben werden (siehe z. B. [29]), auf deren Herleitung hier allerdings verzichtet werden soll. Als Resultat sei festgehalten, dass in diesem Fall der zweite Term in Gleichung (3.5) maximal wird, wenn die gesuchten Parameter μ_j und Σ_j folgende Werte annehmen:

$$\mu_j = \frac{\sum_{k=1}^{K} p(z^k = j|y^k, \Theta^i)\, y^k}{\sum_{k=1}^{K} p(z^k = j|y^k, \Theta^i)} \tag{3.10}$$

$$\Sigma_j = \frac{\sum_{k=1}^{K} p(z^k = j|y^k, \Theta^i)(y^k - \mu_j)(y^k - \mu_j)^T}{\sum_{k=1}^{K} p(z^k = j|y^k, \Theta^i)}. \tag{3.11}$$

Diese Reestimierungsformeln können als Verallgemeinerungen der bekannten statistischen Schätzer für den Mittelwert und die Varianz angesehen werden. Im trivialen Fall nur eines Komponentenmodells gehen sie nämlich in letztere über.

Nach diesem Exkurs der ML-Schätzung in Mischmodellen soll im nächsten Abschnitt die Grundidee dieser Modellierung, ausgedrückt in Gleichung (3.1), aufgegriffen und zur Clusterung von Sequenzen mit Hidden-Markov-Modellen eingesetzt werden.

3.2 MIX-HMM-Algorithmus

Im Abschnitt 2.3.5 wurde ein HMM-Clusteralgorithmus entwickelt, der die Klassifikation von Sequenzen und das Training von Hidden-Markov-Modellen kombiniert. Zu jedem Zeitpunkt der Iteration existiert dort eine eindeutige Zuordnung der Sequenzen zu den Modellen und der Baum-Welch-Algorithmus trainiert die Modelle ausschließlich auf der jeweiligen Sequenzpartition. Eine nahe liegende Alternative, die besonders bei Daten, denen keine klare, natürliche Trennung in Gruppen zugrunde liegt, Anwendung finden kann, besteht darin, jede Sequenz zum Training aller Modelle zu verwenden. Der Einfluss, den eine Sequenz auf die Parameterschätzung eines bestimmten Modells hat, wird dann davon abhängig sein, wie wahrscheinlich es ist, dass die Sequenz von dem betreffenden Modell stammt. Modelle, die sehr „fern“ liegen, sollen entsprechend wenig beim Training von der Sequenz beeinflusst werden, während wahrscheinliche Modelle entsprechend stärker in der Parameteroptimierung von der Sequenz abhängen.

Diese anschauliche Umschreibung soll nun präzisiert werden, indem gezeigt wird, wie der im vorherigen Abschnitt erläuterte Ansatz einer Mischmodellierung auf die Clusterung von Sequenzen mit Hidden-Markov-Modellen angewendet werden kann. Der sich daraus ergebenden Algorithmus bekommt den Namen MIX-HMM.

Erneut seien K unabhängige Sequenzen und C Hidden-Markov-Modelle mit fester Topologie gegeben:

$$\begin{aligned} O &= (O^1, \ldots, O^K), \\ \Lambda &= (\lambda_1, \ldots, \lambda_C). \end{aligned}$$

Wenn $\alpha = (\alpha_1, \ldots, \alpha_C)$ den Vektor der Komponentenwahrscheinlichkeiten bezeichnet und $p_j(O^k|\lambda_j)$ die Wahrscheinlichkeitsdichte der Sequenz O^k im Model λ_j, dann ergibt sich die Wahrscheinlichkeitsdichte einer Sequenz O^k in der HMM-Mischmodellierung zu

$$p(O^k|\Lambda, \alpha) = \sum_{j=1}^{C} \alpha_j p_j(O^k|\lambda_j)$$

und die zu maximierende Log-Likelihood-Funktion hat bei gegebenen Sequenzen O die folgende Form:

$$\log(L_O(\Lambda, \alpha)) = \sum_{k=1}^{K} \log \left(\sum_{j=1}^{C} \alpha_j p_j(O^k|\lambda_j) \right). \tag{3.12}$$

Abgesehen von der speziellen Gestalt der Modellwahrscheinlichkeitsdichte $p_j(O^k|\lambda_j)$ ist dies genau die gleiche Funktion wie beim allgemeinen Mischmodell gegeben durch Gleichung (3.2). Die Maximierung von (3.12) geschieht analog zum vorherigen Abschnitt über den EM-Algorithmus und führt gemäß Gleichung (3.5) im i-ten Schritt auf folgende Q-Funktion:

$$\begin{aligned} Q((\alpha,\Lambda),(\alpha^i,\Lambda^i)) &= \sum_{k=1}^{K}\sum_{j=1}^{C} \log(\alpha_j)\, pr(j|\Lambda^i,\alpha^i,O^k) + \\ &\quad \sum_{k=1}^{K}\sum_{j=1}^{C} \log(p_j(O^k|\lambda_j))\, pr(j|\Lambda^i,\alpha^i,O^k). \end{aligned} \tag{3.13}$$

Hierbei bezeichnet $pr(j|\Lambda^i,\alpha^i,O^k)$ die Wahrscheinlichkeit des Vorliegens von Modell j bei gegebener Sequenz O^k und gegebenen Parametern aller Modelle Λ^i,α^i und gemäß Gleichung (3.8) gilt:

$$pr(j|\Lambda^i,\alpha^i,O^k) = \frac{\alpha_j^i p_j(O^k|\lambda_j^i)}{\sum_{l=1}^{C}\alpha_l^i p_l(O^k|\lambda_l^i)}. \tag{3.14}$$

Rekursive Berechnung von a posteriori Wahrscheinlichkeiten

Eine direkte Berechnung der a posteriori Wahrscheinlichkeit $pr(j|\Lambda^i,\alpha^i,O^k)$ nach Gleichung (3.14) kann zu numerischen Problemen führen, da der Ausdruck $p_j(O^k|\lambda_j)$ zu klein werden kann. Es ist aber möglich, einen rekursiven Algorithmus zur Berechnung dieser Größe anzugeben. Die Darstellung folgt der Vorgehensweise, wie sie in [20] beschrieben wird. Erneut sei mit $O_{(t)}$ folgende Teilsequenz gekennzeichnet:

$$O_{(t)} := O_1 \ldots O_t.$$

Der Sequenzindex k und die Modellparameter Λ^i,α^i werden in den folgenden Gleichungen aus Gründen einer besseren Übersichtlichkeit nicht aufgeführt:

$$pr(j|\Lambda^i,\alpha^i,O^k) \rightarrow pr(j|O).$$

Mit dieser Notation und mit der Bayes-Formel gilt für $pr(j|O_{(t+1)})$ folgende Rekursionsgleichung:

$$\begin{aligned} pr(j|O_{(t+1)}) &= pr(j|O_{(t)},O_{t+1}) \\ &= \frac{p(j,O_{t+1}|O_{(t)})}{p(O_{t+1}|O_{(t)})} \\ &= \frac{p(j,O_{t+1}|O_{(t)})}{\sum_{l=1}^{C} p(l,O_{t+1}|O_{(t)})} \\ &= \frac{p(O_{t+1}|j,O_{(t)})pr(j|O_{(t)})}{\sum_{l=1}^{C} p(O_{t+1}|l,O_{(t)})pr(l|O_{(t)})}. \end{aligned} \tag{3.15}$$

Somit bleibt zur rekursiven Berechnung von $pr(j|O)$ noch zu klären, wie die Größe $p(O_{t+1}|j, O_{(t)})$ berechnet werden kann. Es gilt

$$\begin{aligned} p(O_{t+1}|j, O_{(t)}) &= \frac{p(O_{(t)}, O_{t+1}|j)}{p(O_{(t)}|j)} \\ &= \frac{p(O_{(t+1)}|j)}{p(O_{(t)}|j)} \\ &= \exp\left(\log(p(O_{(t+1)}|j)) - \log(p(O_{(t)}|j))\right). \end{aligned} \tag{3.16}$$

Mit geeigneter Initialisierung durch

$$pr(j|O_{(0)}) := pr(j) = \alpha_j \tag{3.17}$$

definiert Gleichung (3.15) einen rekursiven Algorithmus zur Berechnung der gesuchten a posteriori Wahrscheinlichkeiten der Modelle bei gegebener Sequenz O.

Der Algorithmus MIX-HMM

Im Folgenden wird ein Algorithmus angegeben, der den Ausdruck (3.13) mit jeder Iteration vergrößert oder zumindest nicht verkleinert. Beim MIX-HMM-Algorithmus handelt es sich um einen generalisierten EM-Algorithmus (Bemerkung 2.5.2) und das Verfahren konvergiert gegen einen stationären Wert der Likelihood-Funktion (3.12). Dies wird nach der Vorstellung des Algorithmus bewiesen. Zur Darstellung des Algorithmus wird folgende neue Notation verwendet: Mit $\mathbf{BW}(\lambda_j, O, pr(j|\lambda, \alpha, O^k))$ wird das gewichtete Baum-Welch-Training (s. Abschnitt 4.6) unter Verwendung der Sequenzen $O = (O^1, \ldots, O^K)$ und der Initialparameter λ_j bezeichnet, wobei jede Sequenz O^k mit der Komponentenwahrscheinlichkeit $pr(j|\lambda, \alpha, O^k)$ gewichtet wird. Der Rückgabewert dieser Prozedur ist der gegenüber λ_j verbesserte neue Parameter $\hat{\lambda}_j$.

MIX-HMM-Algorithmus

Eingabe:

Sequenzen $O = (O^1, \ldots, O^K)$
C Initialmodelle $\Lambda^0 = (\lambda_1^0, \ldots, \lambda_C^0)$
Abbruchkriterium ε
maximale Iterationszahl I
(genügend kleiner) Initialfunktionswert *ZMIN*

Initialisierung:

$Z^0 :=$ *ZMIN*
Festlegung von $pr(j|O^k)$
mit $\sum_{j=1}^{C} pr(j|O^k) = 1, \quad pr(j|O^k) \geq 0$ f. a. j, k
Festlegung von α_j^0
mit $\sum_{j=1}^{C} \alpha_j^0 = 1, \quad \alpha_j^0 \geq 0$ f. a. j

for $i := 0$ **to** I **do**
Berechne

a priori Modellwahrscheinlichkeit

$$\alpha_j^{i+1} := \frac{1}{K} \sum_{k=1}^{K} pr(j|\Lambda^i, \alpha^i, O^k), \qquad 1 \leq j \leq C \tag{3.18}$$

Modellparameter

$$\lambda_j^{i+1} := \mathbf{BW}(\lambda_j^i, O, pr(j|\Lambda^i, \alpha^i, O^k)), \qquad 1 \leq j \leq C \tag{3.19}$$

neuen Zielfunktionswert

$$\begin{aligned} Z^{i+1} \quad = \quad & \sum_{k=1}^{K} \sum_{j=1}^{C} \log(\alpha_j^{i+1})\, pr(j|\Lambda^i, \alpha^i, O^k) + \\ & \sum_{k=1}^{K} \sum_{j=1}^{C} \log(p_j(O^k|\lambda_j^{i+1}))\, pr(j|\Lambda^i, \alpha^i, O^k). \end{aligned} \tag{3.20}$$

if ($Z^{i+1} - Z^i < \varepsilon$) **then**
break
else
Berechne $pr(j|\Lambda^{i+1}, \alpha^{i+1}, O^k)$ unter Verwendung
von (3.15), (3.16) und (3.17), f. a. j, k
end if
end for

Ausgabe: Trainierte Modelle $\hat{\Lambda}$, a priori Wahrscheinlichkeiten α_j.

Zur Initialisierung der $pr(j|O^k)$ und zur Wahl der Initialmodelle Λ^0 werden im Abschnitt 5.2 verschiedene Möglichkeiten untersucht.

Durch diesen Algorithmus wird eine Folge von Parametern (α^i, Λ^i) definiert und es gilt der folgende Satz:

Satz 3.2.1 *Gegeben sind eine Menge von Sequenzen $O = (O^1, \ldots, O^K)$ und zwei aus der Anwendung des obigen Algorithmus auf diese Sequenzen sich ergebende, aufeinander folgende Parametervektoren (α^i, Λ^i) und $(\alpha^{i+1}, \Lambda^{i+1})$. Dann gilt für die Log-Likelihood Funktion aus Gleichung* (3.12)

$$\log(L_O(\Lambda^{i+1}, \alpha^{i+1})) \geq \log(L_O(\Lambda^i, \alpha^i)).$$

Beweis
Es genügt zu zeigen, dass für die Q-Funktion nach Gleichung (3.13) gilt:

$$Q((\alpha^{i+1}, \Lambda^{i+1}), (\alpha^i, \Lambda^i)) \geq Q((\alpha^i, \Lambda^i), (\alpha^i, \Lambda^i))$$

da daraus mit Bemerkung 2.5.2 die Behauptung folgt. Wie direkt aus der Gleichung (3.13) ersichtlich, lässt sich die Q-Funktion so in zwei Terme aufspalten, dass der erste Ausdruck nur

von α und der zweite nur von λ abhängt. Daher ist die Behauptung erfüllt, wenn sie für beide Ausdrücke getrennt erfüllt ist. Es ist also zu zeigen, dass gilt

$$\sum_{k=1}^{K}\sum_{j=1}^{C}\log(\alpha_j^{i+1})\,pr(j|\Lambda^i,\alpha^i,O^k) \geq \sum_{k=1}^{K}\sum_{j=1}^{C}\log(\alpha_j^{i})\,pr(j|\Lambda^i,\alpha^i,O^k) \tag{3.21}$$

und

$$\sum_{k=1}^{K}\sum_{j=1}^{C}\log(p_j(O^k|\lambda_j^{i+1}))\,pr(j|\Lambda^i,\alpha^i,O^k) \geq \sum_{k=1}^{K}\sum_{j=1}^{C}\log(p_j(O^k|\lambda_j^{i}))\,pr(j|\Lambda^i,\alpha^i,O^k). \tag{3.22}$$

Die Ungleichung (3.21) wurde schon im vorigen Abschnitt gezeigt, da dort mit Hilfe von Lagrange-Multiplikatoren hergeleitet wurde (Gleichungen (3.6) und (3.7)), dass

$$\sum_{k=1}^{K}\sum_{j=1}^{C}\log(\alpha_j)\,pr(j|\lambda^i,\alpha^i,O^k)$$

gerade durch

$$\alpha_j = \frac{1}{K}\sum_{k=1}^{K} pr(j|\lambda^i,\alpha^i,O^k), \qquad 1 \leq j \leq C$$

maximiert wird, in Übereinstimmung mit Gleichung (3.18) des Algorithmus.

Die Gültigkeit der Ungleichung (3.22) ergibt sich aus der Monotonie des Baum-Welch-Algorithmus beim gewichteten HMM-Training. Der Parameter λ_j^i auf der rechten Seite der Ungleichung (3.22) stellt den Initialparameter für den Baum-Welch-Algorithmus dar; λ_j^{i+1} auf der linken Seite stellt den trainierten Parameter dar und damit gilt:

$$\sum_{k=1}^{K}\log(p_j(O^k|\lambda_j^{i+1}))\,pr(j|\Lambda^i,\alpha^i,O^k) \geq \sum_{k=1}^{K}\log(p_j(O^k|\lambda_j^{i}))\,pr(j|\Lambda^i,\alpha^i,O^k), \qquad 1 \leq j \leq C. \tag{3.23}$$

Damit ist gezeigt, dass die Q-Funktion bei jedem Iterationsschritt gleich bleibt oder wächst und daraus ergibt sich die Behauptung des Satzes. □

Wenn die beiden nachfolgenden Bedingungen erfüllt sind, garantiert Satz 3.2.1 die Konvergenz des MIX-HMM-Algorithmus:

1. Die Log-Likelihood-Funktion ist nach oben beschränkt. Es gilt also für alle (Λ, α), die auftreten können:

$$\log(L_O(\Lambda, \alpha)) < \infty.$$

   Dies wird in der Praxis dadurch erreicht, dass die Varianz der Ausgabedichten nicht unter einen kleinen, vorgegebenen Wert fallen darf. Somit wird verhindert, dass die Ausgabedichte singuläre, unendliche Werte annimmt.

2. Die Parameterinitialisierung geschieht so, dass gilt:

$$\log(L_O(\Lambda^0, \alpha^0)) > -\infty.$$

   Es muss also gewährleistet sein, dass die vorgegebenen Sequenzen von den Initialmodellen abgebildet werden können. Durch eine hinreichend groß gewählte Anfangsvarianz ist dies leicht zu erfüllen.

Zum vorgestellten MIX-HMM-Algorithmus sind noch einige Anmerkungen notwendig:

Kopplung zweier EM-Algorithmen: Beim MIX-HMM-Algorithmus handelt es sich um einen EM-Algorithmus , der zur Verbesserung der Q-Funktion wiederum den EM-Algorithmus in Form der Baum-Welch-Reestimierungsformeln einsetzt. Es werden also zwei EM-Algorithmen miteinander gekoppelt.

Lokales Optimum: Ähnlich wie der partitionierende HMM-Clusteralgorithmus findet auch der MIX-HMM-Algorithmus lediglich ein lokales Maximum der Zielfunktion. Das Ergebnis hängt also von der gewählten Initialisierung ab. Dies betrifft sowohl die Initialparameter (α^0, Λ^0) des Mischmodells als auch die Anfangswahrscheinlichkeiten der Komponenten $pr(j|O^k, \Lambda^0)$. Diese Abhängigkeiten werden im Abschnitt 5.2 genauer untersucht.

Partitionierende Clusterung: Wenn es darum geht, die Trainingssequenzen O^k zu partitionieren, also eindeutig einem HMM λ_i zuzuordnen, so ist dies einfach möglich über

$$i = \arg\max_j pr(j|O^k).$$

A priori Wahrscheinlichkeit: Die Durchführung einer Clusterung mit dem MIX-HMM liefert neben den Parametern der Hidden-Markov-Modelle auch die Komponentenwahrscheinlichkeiten α_i. Bei geeigneter Zusammensetzung der Trainingsmenge können diese Werte direkt als a priori Wahrscheinlichkeiten der einzelnen Modelle interpretiert und bei der Verlängerung bestehender bzw. bei der Generierung neuer Sequenzen berücksichtigt werden.

Laufzeit: Mit den Eingabegrößen
M : Anzahl der Komponentenmodelle
K : Anzahl der Clusterobjekte
N : Anzahl der States in jedem HMM
T : Maximale Länge einer Sequenz
ergibt sich eine Komplexität für einen Reestimierungsschritt von
$\mathcal{O}(KMN^2T)$.

Modifikationen: In der praktischen Anwendung kann es zur Beschleunigung des Verfahrens sinnvoll sein, zu Beginn die Parameter-Estimierung der Hidden-Markov-Modelle bereits nach einigen wenigen Baum-Welch-Schritten abzubrechen und im Algorithmus deutlich vor Erreichen der Konvergenz fortzufahren. Dies hängt damit zusammen, dass in der Regel bei den ersten Baum-Welch-Schritten eine deutliche Verbesserung der Log-Likelihood-Funktion eintritt, die sich in den darauffolgenden Iterationen abschwächt.

3.3 Modellierung zusammengesetzter Daten

Im vorherigen Abschnitt ging es um die Modellierung und Klassifizierung von Sequenzen – also von reellen Werten, denen eine zeitliche Abfolge aufgeprägt ist – mit einem HMM-Mischmodell. In diesem Kapitel soll eine wichtige Erweiterung vorgestellt werden, die in vielen Anwendungen, so auch in der Thematik dieser Arbeit, der Simulation von Bausparkollektiven, eine Bedeutung haben kann.

Ausgangspunkt ist die Beobachtung, dass eine Zeitreihe mit weiteren, für eine Modellierung und Clusterung relevanten Merkmalen verknüpft sein kann, die sich nicht im Zeitverlauf ändern. Es soll jetzt die Frage untersucht werden, wie Daten mit dieser komplexen Struktur, bestehend aus einer Sequenz variabler Länge und einem statischen Vektor modelliert und geclustert werden können. Hierzu werden zwei verschiedene Ansätze vorgestellt. Zunächst wird die Mischmodellierung um eine entsprechend komplexe Dichtefunktion erweitert. Eine ähnliche Vorgehensweise wird auch in [38] und [37] vorgeschlagen. Alternativ dazu wird anschließend die Problematik durch eine geeignete HMM-Topologieerweiterung gelöst.

Jedes zu clusternde Datenobjekt D^k setze sich zusammen aus einer Sequenz O^k und einem statischen Vektor X^k:

$$D = (D^1, \ldots, D^K) = ((O^1, X^1), \ldots, (O^K, X^K)). \tag{3.24}$$

Für den Vektor X^k werden zwei Fälle betrachtet:

1. X^k entstammt einer multivariaten Normalverteilung
2. X^k ist eine eindimensionale diskrete Variable

Sequenz und normalverteilte, multivariate Daten
Es wird die Annahme gemacht, dass X^i einer multivariaten Normalverteilung $\mathcal{N}(X|\mu, \Sigma)$ entstammt und dass die Daten mit folgendem Mischmodell angemessen beschrieben werden, wobei $\Theta = (\Lambda, \alpha, \mu, \Sigma)$ als Kurzschreibweise für die Gesamtheit der Modellparameter verwendet wird:

$$p(D^k|\Theta) = \sum_{j=1}^{C} \alpha_j p_j(O^k|\lambda_j)\mathcal{N}_j(X^k|\mu_j, \Sigma_j). \tag{3.25}$$

Dieser Ansatz impliziert eine statistische Unabhängigkeit von O^k und X^k bei gegebener Clusterzugehörigkeit. Es gibt zwei wichtige Gründe, diese Unabhängigkeit bei der Modellbildung zugrunde zu legen:

1. Die passende Parametrisierung einer Verbundwahrscheinlichkeitsdichte von Sequenz und statischem Vektor kann ohne tiefere Einsicht über den Zusammenhang der beiden Größen schwer angegeben werden. Dieses Wissen ist in der Regel nicht vorhanden, sondern soll gerade mit der Durchführung einer Clusterung erlangt werden.
2. Die Unabhängigkeit ist Grundvoraussetzung für die Trennung der Q-Funktion in separat zu maximierende Teile. Dies wiederum ermöglicht den Einsatz vom Baum-Welch-Algorithmus zur HMM-Parameter-Reestimierung.

Mit dieser vereinfachenden Annahme ist es möglich, lokal optimale Parameter für die mit dem Mischmodell korrespondierende Log-Likelihood-Funktion zu finden, die bei den gegebenen Daten folgende Gestalt annimmt:

$$\log(L_D(\Theta)) = \sum_{k=1}^{K} \log\left(\sum_{j=1}^{C} \alpha_j p_j(O^k|\lambda_j)\mathcal{N}_j(X^k|\mu_j,\Sigma_j)\right). \tag{3.26}$$

Die Maximierung dieser Log-Likelihood-Funktion geschieht wieder unter Verwendung des generalisierten EM-Algorithmus und in Anlehnung an die beiden vorhergehenden Unterkapitel ergibt sich im i-ten Schritt die folgende Q-Funktion, die für den M-Schritt verbessert werden muss:

$$\begin{aligned} Q(\Theta,\Theta^i) \;=\; & \sum_{k=1}^{K}\sum_{j=1}^{C} \log(\alpha_j)\, pr(j|\Theta^i,O^k,X^k) + \\ & \sum_{k=1}^{K}\sum_{j=1}^{C} \log(p_j(O^k|\lambda_j))\, pr(j|\Theta^i,O^k,X^k) + \\ & \sum_{k=1}^{K}\sum_{j=1}^{C} \log(\mathcal{N}_j(X^k|\mu_j,\Sigma_j))\, pr(j|\Theta^i,O^k,X^k). \end{aligned} \tag{3.27}$$

Die Verbesserung dieser Funktion zerfällt in drei bekannte Teilprobleme, die durch die Gleichungen (3.7), (3.10), (3.11) und dem gewichteten Training mit dem Baum-Welch-Algorithmus gelöst werden. Die dort verwendete Komponentenwahrscheinlichkeit $pr(j|\Theta^i, y^k)$ muss nur durch $pr(j|\Theta^i, O^k, X^k)$ ersetzt werden.

Sequenz und diskrete Daten

Die Clusterung der Daten mit Hilfe des Mischmodell-Ansatzes bietet den Vorteil, dass sich auch diskrete Merkmale auf natürliche Art und Weise mit den kontinuierlichen Größen verknüpfen und bei einer Clusterung berücksichtigen lassen. Zur Vereinfachung wird angenommen, dass der statische Vektor X^k aus Gleichung (3.24) nur eindimensional ist und G verschiedene Werte annehmen kann. Es gilt also:

$$X^k \in \{1,\ldots,G\}, \qquad 1 \le k \le K.$$

Es wird weiter angenommen, dass das Merkmal X unabhängig von der Sequenz O ist und dass die Verteilung von X im Komponentenmodell j gegeben ist durch die diskrete Dichtefunktion $f_j(X)$, die eindeutig festgelegt wird durch die Parameter $F = (f_{jg})$, $1 \le j \le C$, $1 \le g \le G$, mit

den Nebenbedingungen

$$\begin{aligned} \sum_{g=1}^{G} f_{jg} &= 1, \qquad 1 \le j \le C, \\ f_{jg} &\ge 0, \qquad 1 \le j \le C,\ 1 \le g \le G. \end{aligned} \tag{3.28}$$

Somit gibt f_{jg} die Wahrscheinlichkeit an, dass X im Komponentenmodell j den Wert g annimmt. Die Wahrscheinlichkeitsdichte des zusammengesetzten Datums im Komponentenmodell j ergibt sich zu

$$p(O^k, X^k|\Lambda, F, j) = p_j(O^k|\lambda_j)\, f_j(X^k), \qquad 1 \le j \le C. \tag{3.29}$$

Mit diesem Ansatz und der Notation $\Theta = (\Lambda, F, \alpha)$ ergibt sich die Log-Likelihood-Funktion zu

$$\log(L_D(\Theta)) = \sum_{k=1}^{K} \log\left(\sum_{j=1}^{C} \alpha_j p_j(O^k|\lambda_j) f_j(X^k)\right) \tag{3.30}$$

und die Q-Funktion zu

$$\begin{aligned} Q(\Theta, \Theta^i) &= \sum_{k=1}^{K}\sum_{j=1}^{C} \log(\alpha_j)\, pr(j|\Theta^i, O^k, X^k) + \\ &\quad \sum_{k=1}^{K}\sum_{j=1}^{C} \log(p_j(O^k|\lambda_j))\, pr(j|\Theta^i, O^k, X^k) + \\ &\quad \sum_{k=1}^{K}\sum_{j=1}^{C} \log(f_j(X^k))\, pr(j|\Theta^i, O^k, X^k). \end{aligned} \tag{3.31}$$

Wieder geht es darum, die Q-Funktion zu verbessern. Die beiden ersten Terme in (3.31) sind gegenüber Gleichung (3.27) unverändert geblieben, sodass nur noch zu klären ist, wie der dritte Ausdruck mit jeder Iteration verbessert oder maximiert werden kann.
Es gilt:

$$\begin{aligned} &\sum_{j=1}^{C}\sum_{k=1}^{K} \log(f_j(X^k))\, pr(j|\Theta^i, O^k, X^k) \\ &= \sum_{j=1}^{C}\sum_{k=1}^{K}\sum_{g=1}^{G} \delta_{kg} \log(f_j(g))\, pr(j|\Theta^i, O^k, X^k) \\ &= \sum_{j=1}^{C}\sum_{g=1}^{G} \log(f_j(g)) \sum_{k=1}^{K} \delta_{kg}\, pr(j|\Theta^i, O^k, X^k) \end{aligned} \tag{3.32}$$

mit

$$\delta_{kg} = \begin{cases} 1 & \text{falls} \quad X^k = g \\ 0 & \text{falls} \quad X^k \neq g. \end{cases}$$

Der Ausdruck in (3.32) hat eine sehr ähnliche Struktur wie der Term für die Komponentengewichte α_j in Gleichung (3.6). Daher kann zur Maximierung wieder die Methode der Lagrange-Multiplikatoren verwendet werden, die für jedes l, m auf folgende Gleichung führt:

$$\frac{\partial}{\partial f_{lm}} \left[\sum_{j=1}^{C} \sum_{g=1}^{G} \log(f_j(g)) \sum_{k=1}^{K} \delta_{kg} pr(j|\Theta^i, O^k, X^k) + \gamma \left(\sum_{g=1}^{G} f_l(g) - 1 \right) \right] = 0. \tag{3.33}$$

Der sich aus der Ableitung ergebende Ausdruck wird nach f_{lm} aufgelöst, mit dem Resultat:

$$f_{lm} = -\frac{1}{\gamma} \sum_{k=1}^{K} \delta_{km} pr(l|\Theta^i, O^k, X^k). \tag{3.34}$$

Durch Summation dieser Gleichung über alle möglichen Werte von m erhält man mit der Nebenbedingung (3.28) und der Gleichung (3.7)

$$\begin{aligned} \gamma &= -\sum_{k=1}^{K} pr(l|\Theta^i, O^k, X^k) \\ &= -K\,\alpha_l. \end{aligned}$$

Dies eingesetzt in Gleichung (3.34) ergibt den optimalen Wert für f_{lm} und damit den Schätzer f_{lm}^{i+1} für die nächste Iteration des EM-Algorithmus:

$$f_{lm}^{i+1} = \frac{1}{K\alpha_l} \sum_{k=1}^{K} \delta_{km} pr(l|\Theta^i, O^k, X^k). \tag{3.35}$$

Wie sich leicht sehen lässt, wird mit dieser Lösung die Nebenbedingung (3.28) erfüllt:

$$\begin{aligned} \sum_{m=1}^{G} f_{lm}^{i+1} &= \frac{1}{K\alpha_l} \sum_{m=1}^{G} \sum_{k=1}^{K} \delta_{km} pr(l|\Theta^i, O^k, X^k) \\ &= \frac{1}{K\alpha_l} \sum_{k=1}^{K} pr(l|\Theta^i, O^k, X^k) \\ &= \frac{\alpha_l}{\alpha_l} = 1. \end{aligned}$$

Somit sind alle Lösungsformeln zur Verbesserung der Q-Funktion, sowohl im Fall von normalverteilten kontinuierlichen Daten, als auch von Größen einer diskreten Verteilung hergeleitet. Ein Algorithmus, der die Maximierung von (3.26) bzw. (3.30) durchführen soll, ergibt sich aus dem MIX-HMM-Algorithmus aus Abschnitt 3.2 dadurch, dass neben den dort beschriebenen Berechnungen in jeder Iteration neue Schätzer für μ, Σ mit den Gleichungen (3.10) und (3.11) bzw. für f_{lm} mit Gleichung (3.35) berechnet werden.

Zur Berechnung der Parameter f_{lm} nach Gleichung (3.35) ist noch zu klären, wie die Komponentenwahrscheinlichkeit $pr(l|\Theta^i, O^k, X^k)$ berechnet werden kann. Die ist möglich mit dem im

Abschnitt 3.2 vorgestellten iterativen Berechnungsverfahren und unter Verwendung der dort eingeführten Notation für Teilsequenzen ergibt sich in Anlehnung an Gleichung (3.15) folgende Rekursion:

$$\begin{aligned} & pr(j|\Theta^i, X^k, O^k_{(t+1)}) \\ &= \frac{p(O^k_{t+1}|j, \Theta^i, X^k, O^k_{(t)})\, pr(j|\Theta^i, X^k, O^k_{(t)})}{\sum_{l=1}^{C} p(O^k_{t+1}|l, \Theta^i, X^k, O^k_{(t)})\, pr(l|\Theta^i, X^k, O^k_{(t)})}. \end{aligned} \tag{3.36}$$

Außerdem gilt:

$$\begin{aligned} & p(O^k_{t+1}|j, \Theta^i, X^k, O^k_{(t)}) \\ &= \frac{p(O^k_{(t+1)}|j, \Theta^i, X^k)}{p(O^k_{(t)}|j, \Theta^i, X^k)} \\ &= \frac{p(O^k_{(t+1)}|j, \Theta^i)}{p(O^k_{(t)}|j, \Theta^i)} \qquad (3.37) \\ &= \exp(\log(p(O^k_{(t+1)}|j, \Theta^i)) - \log(p(O^k_{(t)}|j, \Theta^i))). \qquad (3.38) \end{aligned}$$

Gleichung (3.37) ergibt sich aus der Annahme, dass O^k und X^k bei gegebenem Komponentenmodell statistisch unabhängig sind. In Gleichung (3.38) tauchen die problematischen Wahrscheinlichkeitsdichten ausschließlich im Logarithmus auf und können mit dem Forward-Algorithmus berechnet werden.

Erweiterung der HMM-Topologie

Die Clusterung zusammengesetzter Daten, also Daten bestehend aus einer Sequenz und einem statischen Vektor, kann auch mit Hilfe einer geeigneten Erweiterung der HMM-Topologie durchgeführt werden. Dies stellt eine Alternative zum oben beschriebenen Vorgehen dar.

Beispielhaft sei hier wieder der Fall einer eindimensionalen, diskreten Variablen X betrachtet, die mit der Sequenz verknüpft ist. Weiter wird angenommen, dass diese Variable jeder Sequenz vorangestellt ist, so dass sich eine Gesamtsequenz ergibt, die folgende Gestalt hat:

$$O^k = (X^k, O^k_1, \ldots, O^k_T). \tag{3.39}$$

Der ursprünglichen Modelltopologie zur Abbildung der Sequenz wird ein weiterer Zustand (q_x) hinzugefügt (siehe Abbildung 3.2), der ausschließlich der Ausgabe von X dient.

Da jede Sequenz mit X beginnt, wird das Training des Modells so initialisiert, dass die Startwahrscheinlichkeit für q_x gleich eins und die aller anderen Zustände gleich null ist. Übergänge aus diesem neuen Anfangszustand sind nur in solche Zustände erlaubt, für die im ursprünglichen Modell eine Startwahrscheinlichkeit größer null vorgesehen war, und die entsprechenden Übergangswahrscheinlichkeiten im erweiterten Modell stimmen gerade mit den Startwahrscheinlichkeiten im ursprünglichen Modell überein.

Die Wahrscheinkeitsdichte $p(O^k|\lambda)$ ergibt sich in dieser Modellierung zu

$$\begin{aligned} p(O^k|\lambda) &= \sum_{\text{alle } Q} f_{q_x}(X^k) a_{q_x q_1} f_{q_1}(O^k_1) \prod_{t=2}^{T} a_{q_{t-1} q_t} f_{q_t}(O_t) \\ &= f_{q_x}(X^k) p(O^k_1, \ldots, O^k_T|\lambda). \end{aligned} \tag{3.40}$$

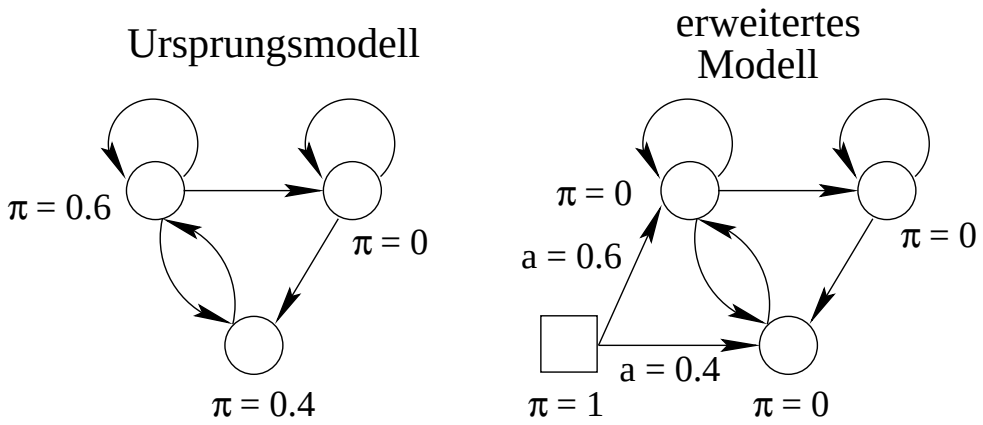

Abbildung 3.2: *Abbildung eines diskreten, zusätzlichen Merkmals durch eine Modellerweiterung. Der weitere Zustand (Quadrat) dient der Ausgabe von X.*

In Gleichung (3.40) kennzeichnet $f_{q_x}(X)$ die diskrete Dichtefunktion des Zustandes q_x zur Ausgabe von X und der obige Ausdruck ist identisch mit der Wahrscheinlichkeitsdichte einer Komponente im Mischmodell nach Gleichung (3.29) zur Ausgabe des zusammengesetzten Datums $O^k = (X^k, O_1^k, \ldots, O_T^k)$. Somit kann die Abbildung eines zusätzlichen, diskreten Merkmals durch die vorgeschlagene Modellerweiterung bewirkt werden. Die Verknüpfung mit weiteren Merkmalen oder mit vektoriellen Merkmalen ist auf ganz analoge Weise möglich.

In diesem Abschnitt wurde also gezeigt, wie sich die Kombination einer Sequenz mit einem normalverteilten Vektor oder einem diskreten Vektor clustern lässt. Dies ist möglich durch eine Berücksichtigung der zusätzlichen Merkmale in den Wahrscheinlichkeitsdichten der Komponenten des Mischmodells (Gleichungen (3.25) und (3.29)) oder durch eine Erweiterung der Modelltopologie des Hidden-Markov-Modells wie oben erläutert. Das beschriebene Verfahren kann einfach auf kompliziertere Kombinationen auch mehrerer Sequenzen und statischer Vektoren erweitert werden, wenn es bei gegebener Komponentenmodellzugehörigkeit gerechtfertigt ist, die verschiedenen Merkmale bzw. Sequenzen als voneinander unabhängig zu betrachten, da dann die notwendige Maximierung der dazugehörigen Q-Funktion in leichtere Subprobleme unterteilt werden kann.

Wenn diese Unabhängigkeit nicht zu rechtfertigen ist, hängt die Durchführbarkeit der MIX-Clusterung im Wesentlichen von zwei Punkten ab:

1. Kann eine angemessene Parametrisierung der Gesamtwahrscheinlichkeitsdichte der Merkmale angegeben werden?

2. Ist es möglich, die mit obiger Parametrisierung korrespondierende Q-Funktion zu maximieren bzw. sicherzustellen, dass die Q-Funktion mit jeder Iteration wächst?

3.4 Wahrscheinlichkeit und Abstandsmaß

Mit der beschriebenen Vorgehensweise ist es möglich, Objekte zu clustern, die sich durch die Abbildung auf einen einfachen Vektor und den Einsatz eines herkömmlichen geometrischen Clusterverfahrens nicht angemessen beschreiben lassen. Das gilt für einfache Zeitreihen, da mit der Abbildung auf einen Vektor die zeitliche Abfolge der Merkmale verloren geht. Außerdem führt dies bei unterschiedlich langen Zeitreihen zu Vektoren unterschiedlicher Dimension, was bei herkömmlichen Clusterverfahren große Probleme bereitet. In verstärktem Maße gilt es für Objekte, die sich aus zeitabhängigen und zeitunabhängigen Merkmalen zusammensetzen, die zudem noch aus einer kontinuierlichen oder diskreten Grundgesamtheit stammen können. Die Abbildung 3.3 veranschaulicht diese Problematik am einfachen Beispiel dreier Zeitreihen.

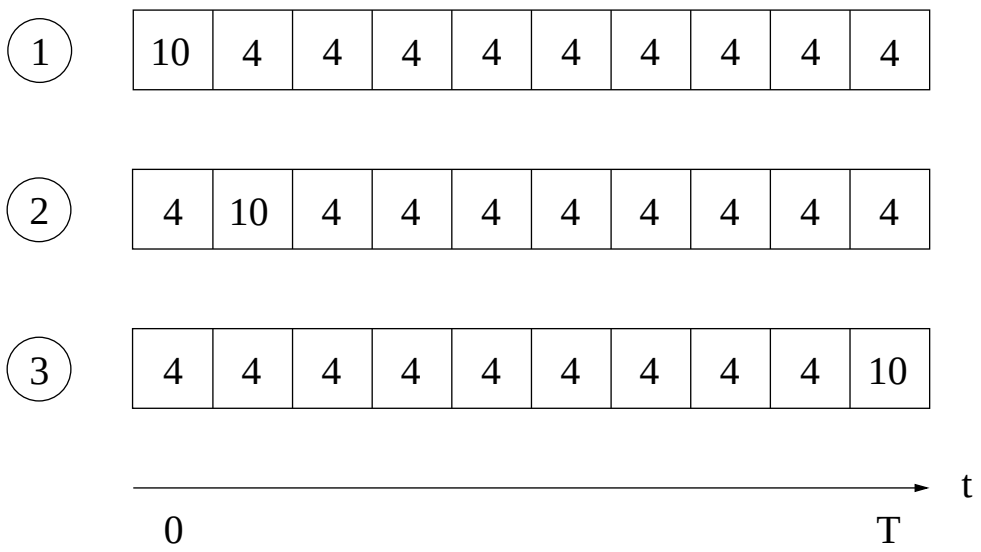

Abbildung 3.3: Drei Sequenzen mit gleichen Abständen

Bei einem geometrischen Clusterverfahren, beispielsweise mit einem quadratischen Abstandsmaß,

$$d(x,y) := \sum_{t=1}^{T} (x_t - y_t)^2, \tag{3.41}$$

ergibt sich für je zwei dieser Sequenzen immer der gleiche Abstand von 72. In den meisten Anwendungen, so auch in der Bausparmodellierung, wird es jedoch angemessener sein, Sequenz Nr. 1 und Nr. 2 als ähnlicher anzusehen als beispielsweise Sequenz Nr. 1 und Sequenz Nr. 3. Anschaulich begründet liegt die größere Ähnlichkeit von Reihe Nr. 1 und Reihe Nr. 2 darin, dass das herausstechende Ereignis, symbolisiert durch die 10, bei diesen Sequenzen zeitnaher ist als bei den übrigen Paaren.

Bei einem geometrischen Clusterverfahren stellt sich somit bei diesen Daten das fundamentale Problem, ein geeignetes Abstandsmaß zu finden. In dem hier beschriebenen Verfahren ist es

nicht notwendig, einen Abstand $d(D^i, D^k)$ zwischen den Clusterobjekten zu definieren, da zwei Objekte nicht direkt miteinander verglichen werden müssen. Stattdessen müssen für jedes Objekt D^i und für jedes Modell l, festgelegt durch den Parametervektor Θ_l, die beiden folgenden Größen berechenbar sein:

Daten-Wahrscheinlichkeit(sdichte): $p(D^i|l, \Theta_l)$

Komponentenmodell-Wahrscheinlichkeit: $pr(l|D^i, \Theta_l)$.

Die Größe $pr(l|D^i, \Theta_l)$ ist in der Mischmodellierung ein Maß dafür, wie groß der Einfluss des Objektes D^i auf die Parameterschätzung von Modell l ist. Je größer die Wahrscheinlichkeit ist, dass D^i vom Modell l stammt, desto größer wird auch der Einfluss, den das Datum auf die Modellparameter hat. Über diese Größe wird das Objekt gewissermaßen auf die verschiedenen Modelle aufgeteilt. Der Ausdruck $p(D^i|l, \Theta_l)$ ist natürlich relevant, da er direkt in die Berechnung der Zielfunktion eingeht.

Kapitel 4

Hidden-Markov-Bauspar-Modell (HMBM)

In diesem Kapitel wird erläutert, wie die in den beiden vorangegangenen Kapiteln beschriebenen Hidden-Markov-Modelle zur Modellierung von Bausparkollektiven eingesetzt werden können. Dazu werden hier die Grundelemente des HMBM vorgestellt. Den beiden wichtigen Themenbereichen „Training der Modelle“ und „Durchführung von Simulationen“ sind jeweils eigene Kapitel gewidmet (Kapitel 5 und Kapitel 6).

Im ersten Abschnitt dieses Kapitels werden die für das Verständnis der Arbeit notwendigen Begriffe des Bausparens und der Sinn und Zweck von Bausparsimulationen erläutert. Außerdem werden einige bestehende alternative Modelle zur Simulation von Bausparkollektiven vorgestellt. Daran anschließend wird eine Übersicht des Modellierungsansatzes mit dem HMBM gegeben. Im dritten Unterkapitel wird erklärt, wie aus den Bauspardaten Sequenzen erzeugt werden. Zusätzlich werden die in dieser Arbeit eingesetzten Mengen von Sequenzen beschrieben. Sehr entscheidenden Einfluss auf die Simulationsergebnisse hat die Wahl der Modelltopologie. Was hierbei von Bedeutung ist, wird im vierten Abschnitt erörtert. Die beiden letzten Abschnitte widmen sich zwei wichtigen Erweiterungen der HMM-Modellierung: der geeigneten Definition der Übergangsklassen und der Rolle der Bausparsumme.

4.1 Bausparkassen-Modelle

4.1.1 Grundbegriffe Bausparen

In diesem Abschnitt sollen die wesentlichen Begriffe und die Idee des Bausparens erläutert werden. Eine ausführlichere Übersicht zum Thema Bausparen findet sich z. B. in [4, 26]. Die generelle Idee eines Bausparvertrages besteht darin, dass mit Sparzahlungen, die über einen gewissen Zeitraum geleistet werden, ein Anspruch auf ein zinsgünstiges Darlehen erworben wird. Der Ablauf eines typischen Bausparvertrages gliedert sich in drei Phasen: In der ersten Phase, der Sparphase, leistet der Bausparer Sparzahlungen auf sein Bausparkonto und baut somit ein

Guthaben auf. Die zweite Phase (Zuteilungsphase) beginnt mit der Zuteilung und endet im Fall der Darlehensnahme mit der Darlehensauszahlung. In ihr wird dem Bausparer das angesparte Guthaben einschließlich der angesammelten Sparzinsen und gegebenenfalls ein zinsgünstiges Darlehen ausgezahlt. Während der dritten Phase wird das ausgezahlte Darlehen vom Bausparer getilgt. Abbildung 4.1 veranschaulicht diese drei Phasen anhand von Guthabens- und Darlehensverläufen eines Vertrages, der gleichmäßig bespart und getilgt wird. Es ist möglich, dass dieser Ablauf durch Kündigung oder Darlehensverzicht verkürzt wird. Die Konvexität der Guthabens- und Darlehenskurve ergibt sich aus den Spar- und Darlehenszinsen.

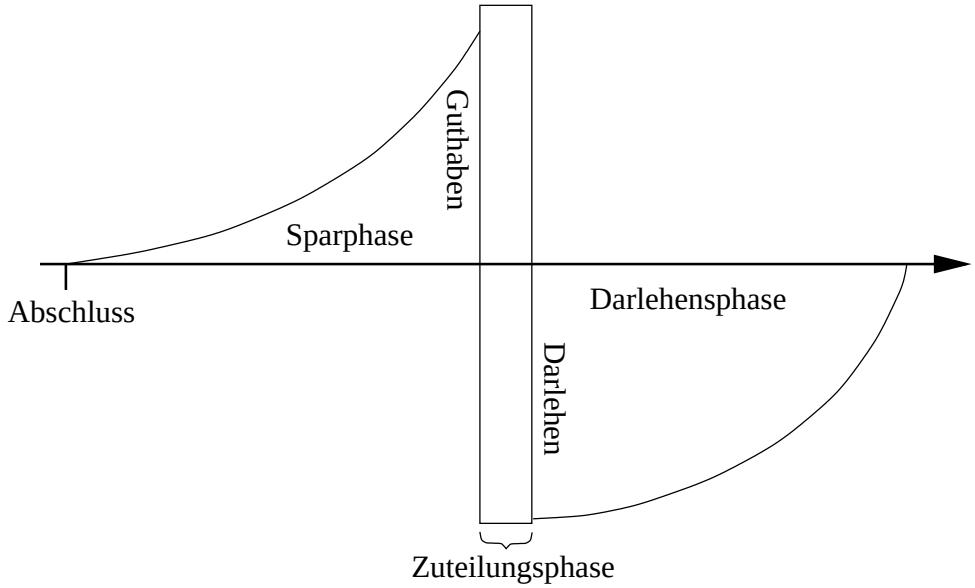

Abbildung 4.1: Die drei Phasen eines Bausparvertrages: Sparphase, Zuteilungsphase und Darlehensphase.

Jeder traditionelle Bausparvertrag wird in einem bestimmten Tarif, in dem alle für den Ablauf eines Vertrages relevanten Parameter festgelegt sind, und über eine bestimmte Bausparsumme, die ein Vielfaches von 1000 DM sein muss, abgeschlossen. Die Bausparsumme hat eine große Bedeutung, da sich aus ihr und dem angesparten Guthaben die Höhe des ausgezahlten Darlehens ergibt. Ohne Berücksichtigung von Gebühren gilt:

$$\text{Darlehen} = \text{Bausparsumme} - \text{Guthaben}.$$

Damit ein Bausparvertrag zugeteilt und damit die Sparphase beendet werden kann, muss er bestimmte, vom jeweiligen Tarif abhängige Mindestbedingungen erfüllen, die nachfolgend erläutert werden:

Mindestspardauer: In den meisten Tarifen ist eine Mindestspardauer von 18 Monaten vorgesehen, vorher ist keine Zuteilung möglich.

Mindestanspargrad: Der Anspargrad eines Bausparvertrages ergibt sich aus dem Quotienten von Guthaben und Bausparsumme. Damit ein Vertrag zugeteilt wird, muss je nach Tarif ein Anspargrad von 40% bzw. 50% vorliegen.

Bewertungszahl (BWZ): Der Vertrag muss eine gewisse Bewertungszahl überschritten haben, um zugeteilt zu werden. Die Bewertungszahl ist ein Maß für die Leistung, die ein Kunde durch seine Sparzahlungen der Gesamtheit der Bausparer (Bausparkollektiv) zur Verfügung gestellt hat. In der Praxis der Bausparkassen werden verschiedene Berechnungsverfahren für die BWZ verwendet. Hierauf und auf die Bedeutung der BWZ für eine HMM-Simulation wird im Abschnitt 4.5 eingegangen.

Jeder Tarif gibt eine bestimmte Regelsparrate vor, also eine Festlegung der monatlichen Sparrate, ausgedrückt durch einen festen Anteil bezogen auf die Bausparsumme. Dies ist jedoch nur ein Richtwert und der Kunde kann jederzeit mit seinen Sparzahlungen von dieser Vorgabe nach unten oder oben abweichen. Hiervon wird in der Realität häufig Gebrauch gemacht; extreme Sparverläufe, wie das Ausbleiben jeglicher Sparzahlungen oder die einmalige Einzahlung zum direkten Erreichen des Mindestanspargrades werden häufig beobachtet. Diese Variabilität der Spareinzahlungen stellt einen zentralen Gegenstand aller Datenanalysen und Simulationen bezogen auf Bausparkollektive dar. In der Darlehensphase ist die Variabilität der Tilgungszahlungen eingeschränkt. Es ist vorgeschrieben, dass der Vertrag mit einer festen, vom gewählten Tarif abhängigen Regeltilgungsrate getilgt wird. Ein Überschreiten der tariflichen Tilgungsrate durch Sondertilgungen wird in den Regel von der Bausparkasse akzeptiert, ein Unterschreiten ist jedoch nicht möglich.

Neben den erwähnten Spar- und Tilgungszahlungen werden in dieser Arbeit eine Reihe weiterer Aktionsmöglichkeiten von Bausparern betrachtet und modelliert, die eine Abweichung vom oben skizzierten typischen Vertragsablauf bewirken. Dies ist von sehr großer Bedeutung, da die folgenden Aktionen ganz erheblichen Einfluss auf die Entwicklung wichtiger Kollektivzeitreihen haben:

Kündigung: Ein Vertrag kann zu jedem Zeitpunkt der Sparphase gekündigt werden. Der Kunde erhält das angesparte Guthaben einschließlich der Sparzinsen ausbezahlt. Es kann zwischen Kündigungen innnerhalb und außerhalb der Sperrfrist unterschieden werden. Bei Kündigungen innerhalb der Sperrfrist verliert der Kunde den Anspruch auf eine eventuelle Wohnungsbauprämie. Bei Kündigung nach Ablauf der Sperrfrist ist dies nicht der Fall. Eine Kündigung nach Erreichen der Zuteilung ist nicht möglich.

Fortsetzung: Ein Bausparer, der die Zuteilung erhalten könnte, also alle oben erwähnten Zuteilungsvoraussetzungen erfüllt, aber die Zuteilung hinauszögert, etwa weil er das Darlehen erst später benötigt, wird als Fortsetzer bezeichnet. Die Kollektivgrößen, die sich auf Fortsetzer beziehen, werden von den Bausparkassen getrennt geführt, da Fortsetzer ein besonderes Risiko für das Bausparkollektiv darstellen. Sie können jederzeit auf eine Zuteilung und Auszahlung bestehen, was bedeutet, dass die Bausparkasse die entsprechenden Kapitalmittel bereithalten muss.

Darlehensverzicht: Wenn nach der Zuteilung vom Bausparer lediglich die Auszahlung des Guthabens und der Zinsen und nicht die des Darlehens gewünscht wird, so spricht man von Darlehensverzicht. Verträge, die auf ihr Darlehen verzichten, können in einzelnen Tarifen einen bedeutenden Anteil erreichen. Es gibt spezielle Tarife, die den Darlehensverzicht fördern; hier liegt also die Motivation zum Abschluss eines Bausparvertrages für den Kunden in einer attraktiven Rendite während der Sparphase und nicht so sehr in einem zinsgünstigen Darlehen.

Verzögerte Auszahlung: Es kommt häufig vor, dass nach der Zuteilung das Darlehen nicht unmittelbar in Anspruch genommen wird. Der Vertrag kann mehrere Jahre in der Zuteilungsphase verharren, ohne in die Darlehensphase einzutreten. Der Unterschied zur Fortsetzung besteht darin, dass ein fortgesetzter Vertrag noch nicht zugeteilt ist und somit eindeutig der Sparphase zuzurechnen ist.

Diese Aktionen werden vom HMBM, so wie es in dieser Arbeit entwickelt wird, abgebildet. Wie dies genau geschieht ist im Abschnitt 4.3.1 beschrieben. Daneben gibt es eine Reihe weiterer Aktionsmöglichkeiten von Bausparern, von denen die wichtigsten nachfolgend erläutert werden.

Der Bausparer hat die Möglichkeit, während der Sparphase die Höhe der Bausparsumme zu verändern (Erhöhung oder Ermäßigung). Damit kann er seinen Bausparvertrag an veränderte persönliche Ziele anpassen. Die Ermäßigung ist eine Möglichkeit, eine vorzeitige Zuteilung zu bewirken. Durch die geringere Bausparsumme erhöhen sich Anspargrad und BWZ des Vertrages und liegen bei geeigneter Wahl der Ermäßigung oberhalb der Mindestvoraussetzungen. Der Preis für die vorgezogene Zuteilung ist natürlich ein verringertes Anfangsdarlehen. Eine weitere Möglichkeit, den Ablauf des Vertrages zu beeinflussen, liegt darin, dass ein Tarifwechsel vorgenommen werden kann. Auch hierdurch ist eine Anpassung an geänderte Sparziele möglich. Falls ein Bausparer mehrere Verträge besitzt, so hat er die Option, diese Verträge zu einem großen Vertrag zu vereinen. Daneben wurden von verschiedenen Bausparkassen so genannte Opti-Modelle entwickelt, die aus einer Kombination von mehreren, genau aufeinander abgestimmten Bausparverträgen bestehen. Die Modelle werden in Verbindung mit Vorfinanzierungskrediten angeboten. Somit stellen sie eine Erweiterung zur klassischen Idee des Bausparens dar, da der Kunde sofort, ohne Sparleistungen gezahlt zu haben, ein Darlehen bekommt. Diese Aktionsmöglichkeiten bleiben in dieser ersten Stufe des HMBM unberücksichtigt, könnten aber bei einer Weiterentwicklung des Modells noch integriert werden.

4.1.2 Simulationen von Bausparkollektiven

Zweck von Simulationsmodellen im Bausparwesen

Die Simulation eines Bausparkollektivs ist aus Sicht der betreffenden Bausparkasse aus verschiedenen Gründen interessant, von denen die wichtigsten nachfolgend erläutert werden.

Ein wichtiger Aspekt ist die Unterstützung der Ertrags- und Liquiditätsplanung durch eine Simulation. In einer sehr vereinfachten Sichtweise kann eine Bausparkasse als geschlossenes System betrachtet werden, in dem Darlehensauszahlungen durch entsprechende Einzahlungen aus der Sparphase kompensiert werden. In der Praxis ist die Situation wesentlich komplizierter, da der Kunde eine beachtliche Wahlfreiheit bezüglich seiner Verhaltensweisen hat. Dies hat zum einen zur Folge, dass die zukünftige Entwicklung relevanter Kollektivgrößen nicht durch den aktuellen Zustand des Kollektivs determiniert ist, und zum anderen kann es dazu führen, dass sich Einnahmen und Aufwendungen der Bausparkasse nicht die Waage halten. Aus dem zweiten Punkt erwächst die Notwendigkeit, liquide Mittel am Markt anzulegen oder Gelder zur Finanzierung von Darlehensauszahlungen aufzunehmen. Ziel einer Simulation ist in diesem Zusammenhang, die Planung solcher Maßnahmen durch Aussagen zur zukünftigen Kollektiventwicklung zu unterstützen.

Ein weiterer Punkt liegt in der Entwicklung neuer Tarife oder in der Veränderung bestehender Tarifparameter. Das Bausparwesen ist ständigen Veränderungen unterworfen, die z. T. aus Änderungen in der Gesetzgebung resultieren und z. T. aus der Konkurrenzsituation zu anderen Bausparkassen erwachsen. Modifizierungen am Tarifgefüge haben immer Auswirkungen auf die zukünftige Entwicklung der Bausparkasse und ein geeignetes Simulationsmodell bietet die Möglichkeit, diese Auswirkungen unter Durchführung von Szenariorechnungen genauer zu untersuchen. Somit kann ein Simulationsmodell Aufschluss geben über die komplexen Zusammenhänge von Tarifparametern und die Entwicklung relevanter Kollektivzeitreihen. Es ist durch Szenariorechnungen auch möglich, Auswirkungen von Veränderungen externer Größen wie z. B. des generellen Zinsniveaus auf das Kollektiv zu untersuchen.

Die Entwicklung eines aussagekräftigen Simulationsmodells setzt eine eingehende Datenanalyse voraus. Wenn das Modell auf Einzelvertragsdaten der Bausparkasse basiert, dann kann diese Analyse sozusagen als Nebeneffekt die in realen Daten immer vorhandenen Datenfehler aufspüren und damit die Datenpflege von Seiten der Bausparkasse unterstützen.

Nachfolgend werden drei in der Vergangenheit entwickelte Modelle zur Simulation von Bausparkollektiven beschrieben.

Schichtenmodell
Das so genannte Schichtenmodell wird in [12] dargestellt. Es basiert darauf, dass das Verhalten von Bausparern eines Kollektivs durch eine geringe Zahl so genannter Schichten abgebildet werden kann. Die Schicht repräsentiert ein ganz bestimmtes, deterministisches Verhaltensmuster in der Spar- und Darlehensphase. Ihre Definition ergibt sich aus Erfahrungswerten der Bausparkassen bezüglich verschiedener relevanter Verhaltensweisen von Bausparern und ist nicht das Ergebnis einer auf realen Einzelverträgen basierenden Datenanalyse. Die Anteile der verschiedenen Schichten am Kollektiv werden in diesem Modell mit einem nichtlinearen Optimierungsverfahren bestimmt. Das Ziel der Optimierung besteht darin, vorgegebene kollektive Eckdaten der Vergangenheit durch eine geeignete Schichtenzusammensetzung möglichst gut zu treffen. Daten einzelner Verträge gehen in dieses Modell nicht ein, lediglich Kollektivgrößen finden Berücksichtigung. Das Schichtenmodell ist geeignet, die Zusammenhänge im so genannten Beharrungszustand, der dadurch charakterisiert ist, dass alle Kollektivgrößen konstant sind, zu untersuchen. Die Modellierung ist damit recht realitätsfern, da dynamische Aspekte nicht abbildbar sind.

Mikrosimulationsmodell
Das Mikrosimulationsmodell wurde im Rahmen zweier Dissertationen entwickelt [21, 41]. Es handelt sich um ein stochastisches Modell, das auf den Einzelvertragsdaten einer Bausparkasse basiert. In diesem Modell werden die Verträge anhand bestimmter Merkmale, die in einem in der Vergangenheit liegenden Referenzjahr ermittelt werden, in Gruppen eingeteilt. Innerhalb dieser Gruppen werden Häufigkeitsverteilungen für verschiedene Aktionen ermittelt. Eine Simulation besteht hier darin, alle vorhandenen Verträge zu jedem Simulationszeitpunkt neu in diese Gruppen einzuteilen und die relevanten Aktionen jedes Vertrages gemäß der Häufigkeiten der Gruppe per Zufallszahlengenerator festzulegen. Auf die Art und Weise wird die zukünftige Entwicklung sowohl bestehender Altverträge als auch neu hinzukommender Verträge generiert. Eine Kollektivsimulation ergibt sich aus der Summation aller Einzelverträge zu jedem Simulationszeitpunkt. Zur Ermittlung der erwähnten Gruppen wurden ursprünglich starre Rasterver-

fahren und in der Weiterentwicklung flexiblere Clusterverfahren eingesetzt. Ein wesentliches Charakteristikum dieses Modells liegt darin, dass zur Gruppeneinteilung jeweils nur Daten eines einzigen Zeitraumes genutzt werden. Der Zeitverlauf eines Vertrages findet keinen direkten Eingang.

Mesoskopisches Modell
Das mesoskopische Modell befindet sich seit etwa vier Jahren im praktischen Einsatz; mit ihm wurden Simulationen für verschiedene Bausparkassen durchgeführt. Es wird ausführlich in [24] beschrieben und greift Ideen aus den beiden vorher erwähnten Modellen auf.

Die Parallele zum Schichtenmodell besteht darin, dass es sich ebenfalls um ein deterministisches Modell handelt. Auch der Grundgedanke, das Kollektiv durch typische Verhaltensweisen abzubilden, findet sich hier wieder. Nur dass diese Verhaltensmuster, im Folgenden auch Prototypen genannt, nicht starr vorgegeben, sondern aus den zur Verfügung stehenden Daten ermittelt werden. Die Gemeinsamkeiten mit dem Mikrosimulationsmodell liegen in der gleichen Datenbasis (auch das mesoskopische Modell verwendet Einzelvertragsdaten) und in der Tatsache, dass auch hier die Clusterung von Daten ein zentraler Bestandteil des Modells ist.

Das mesoskopische Modell kann grob wie folgt beschrieben werden: Aus dem zur Verfügung stehenden Datenbestand wird eine Gruppe von Verträgen mit abgeschlossener Sparphase ausgewählt, die die Struktur der typischen Verhaltensmuster gut abdeckt. Diese Verträge werden mit dem K-Means-Verfahren [13] geclustert, wobei die Spargeldeingänge im Zeitverlauf als Clustermerkmale dienen. Als Abstandsmaß wird hierbei die Summe der quadratischen Spargeldeingangsdifferenzen verwendet, summiert über alle vorhandenen Zeiträume. Das Ergebnis der Clusterung sind deterministische Prototypen. Der Prototyp enthält genaue Angaben zum Sparverlauf bis zum Zeitpunkt der Zuteilung.

Jeder Prototyp wird weiter unterteilt in so genannte Schichten. Eine Schicht enthält alle Verhaltensweisen vom Abschluss bis zur Abwicklung des Vertrages. Das Sparverhalten wird z. B. ergänzt um zusätzliche Angaben zur Darlehensphase oder auch zur Kündigung oder zum Darlehensverzicht. Aus statistischen Untersuchungen von Vergangenheitsdaten werden die Anteile der einzelnen Schichten bestimmt.

Zur Simulation mit dem mesoskopischen Modell werden alle zum Simulationsbeginn aktiven Verträge jahrgangsweise den Prototypen zugeordnet. Ein Vertrag wird als aktiv bezeichnet, wenn er sich in einer der in Abschnitt 4.1.1 erwähnten Phasen (Sparphase, Zuteilungsphase oder Darlehensphase) befindet. Die Zuordnung geschieht mit dem in der Clusterung verwendeten Abstandsmaß unter Einhaltung gewisser Ober- und Unterschranken für die verschiedenen Prototypen. Die Zuordnungsaufgabe wird als Netzwerkfluss-Problem formuliert und mit einem Min-Cost-Flow-Algorithmus gelöst. Nach der Zuordnung werden keine einzelnen Bausparverträge mehr betrachtet, sondern nur noch Bausparsummen und Summen der zugehörigen relevanten Bauspargrößen. Die Summen werden auf die einzelnen Schichten aufgeteilt und jede Schicht wird ab dem Simulationbeginn mit einem Bauspar-Rechentool (NBI) deterministisch weitergerechnet. Die Kollektivsimulation ergibt sich wieder durch Summation über alle Schichten, einschließlich neu gestarteter Schichten, die das zukünftige Neugeschäft abbilden.

Ein Problem des mesoskopischen Modells besteht in der gemeinsamen Clusterung von Verträgen, die aufgrund verschieden langer Sparphasen unterschiedliche Dimensionen besitzen. Zur

Lösung muss für unterschiedlich lange Zeitreihen ein geeignetes Abstandsmaß definiert werden. Details dieser Problematik finden sich in [24].

4.2 Übersicht: Modellierungsansatz

In diesem Abschnitt werden die wesentlichen Elemente und der Ablauf einer vollständigen Simulation eines Bausparkollektivs mit dem HMBM beschrieben. Diese Beschreibung bleibt recht grob, da sie der Übersicht dient und zeigen soll, wie die verschiedenen Modellkomponenten zusammenwirken (s. auch Abbildung 4.2). Eine detailliertere Darstellung folgt in den nächsten Abschnitten, auf die bei den einzelnen Punkten verwiesen wird. Das HMBM gliedert sich in die beiden Bereiche Training der Hidden-Markov-Modelle und Simulation der zukünftigen Sequenzen bzw. der Kollektivzeitreihen. Das Training umfasst die folgenden Punkte:

Training

1. Trainingssequenzen: Auswahl einer geeigneten Traininsmenge und Erzeugung der Trainingssequenzen aus den Bauspardaten (4.3.2, 4.3.1).
2. Clusterung und Training: Unter Vorgabe von Initialmodellen kombinierte Clusterung der Trainingsdaten und Training der Hidden-Markov-Modelle (5).
3. Bestimmung von a priori Wahrscheinlichkeiten der trainierten Modelle (5.2.3).

Die Clusterung der Trainingssequenzen kann entweder mit dem im Abschnitt 2.3.5 beschriebenen HMM-Clusteralgorithmus oder durch die Modellierung als Mischmodell (MIX-HMM) aus Abschnitt 3.2 geschehen.

Simulation
Die trainierten Hidden-Markov-Modelle werden als Generator für neue Zeitreihen und für die Vervollständigung bestehender Sequenzen eingesetzt, woraus sich die Simulation der zukünftigen Kollektiventwicklung ergibt. Die Durchführung einer Simulation setzt sich aus den folgenden Punkten zusammen:

1. Bildung des Gesamtkollektivs aus den Bauspardaten und Klassifikation aller darin enthaltenen Sequenzen.
2. Generierung der Sequenzverlängerungen: Jede Sequenz des Bestandes wird unter Verwendung des aus der Klassifikation zugewiesenen Modells verlängert, bis ein gültiges Endsymbol der Sequenz erreicht ist. (6.2).
3. Erzeugung von Neugeschäft: Unter Verwendung der a priori Modellwahrscheinlichkeiten und geeigneter Volumenvorgaben für die verschiedenen Abschlussjahrgänge werden jahrgangsweise neue Sequenzen generiert (6.2).
4. Kollektivbildung: Durch Summation aller generierten Sequenzen (Bestandsverlängerung und Neugeschäft) wird das Gesamtkollektiv gebildet (6.3).

Bei der Verwendung des MAW-Klassifikators müssen a priori Wahrscheinlichkeiten $pr(j)$ der einzelnen Modelle spezifiziert werden. Dies wird durch den gestrichelten Pfeil in Abbildung 4.2 symbolisiert.

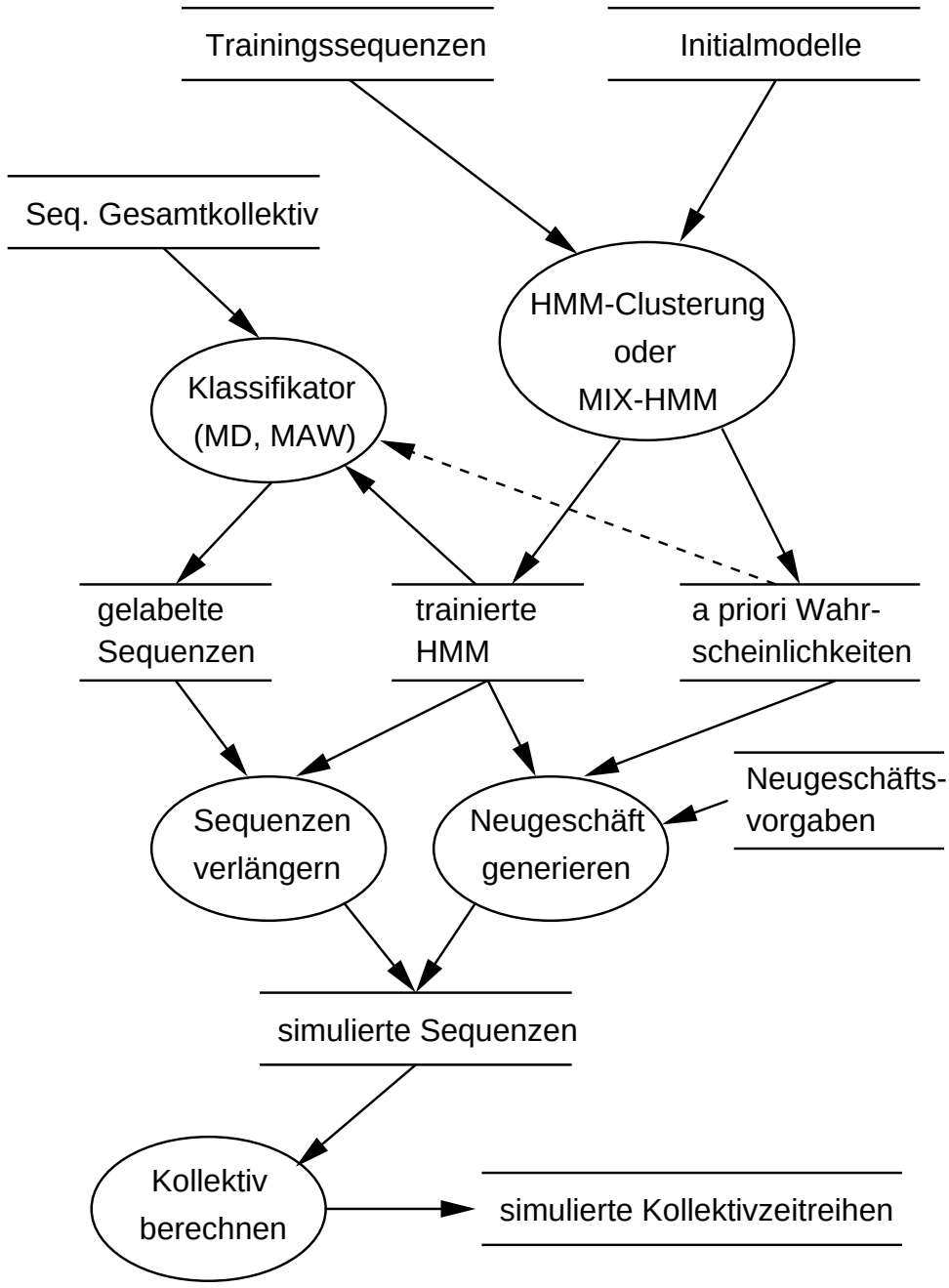

Abbildung 4.2: *Schematische Übersichter der Daten und Prozeduren des HMBM*

4.3 Daten und Aktionen

4.3.1 Bildung von Sequenzen

Die von den Bausparkassen gelieferten Vertragsdaten bestehen aus ca. 50 verschiedenen Merkmalen. Dabei kann unterschieden werden zwischen Merkmalen, die zeitunabhängig sind, wie

z. B. demographischen Daten des Bausparers, und Merkmalen, die sich auf einen bestimmten Zeitraum bzw. einen bestimmten Zeitpunkt beziehen. Merkmale, die sich auf einen Zeitraum beziehen, werden auch als Flussgrößen bezeichnet, während Angaben zu einem festen Zeitpunkt Bestandsgrößen genannt werden. Flussgrößen sind beispielsweise Spar- und Tilgungszahlungen, wohingegen das Guthaben eine Bestandsgröße darstellt. Bei den in dieser Arbeit eingesetzten Daten handelt es sich um Jahresdaten. Das bedeutet, dass die Zeiträume der Flussgrößen jeweils ein Jahr umfassen und dass auch die Bestandsgrößen einmal pro Jahr erfasst werden. Entsprechend bezieht sich auch jedes Symbol einer Sequenz auf genau ein Jahr.

Bei der Bildung der Sequenzen aus der Gesamtmenge der zur Verfügung stehenden Daten müssen verschiedene Fragen beantwortet werden:

1. Welche Merkmale sind für die Modellierung relevant und müssen daher berücksichtigt werden?
2. Ist es hinreichend, eine skalare Zeitreihe zu verwenden oder muss die Abbildung durch eine vektorielle Zeitreihe geschehen?
3. Sollte bei kontinuierlichen Daten auch eine Zeitreihe bestehend aus Symbolen einer kontinuierlichen Gesamtheit verwendet werden oder kann eine Diskretisierung durch Abbildung auf Intervalle vorgenommen werden?

Für das in dieser Arbeit entwickelte Modell HMBM lassen sich diese Fragen wie folgt beantworten:

Zu 1.: Bei der Wahl der Merkmale werden die primären Aktionen des Bausparers berücksichtigt. Hierunter fallen Spar- und Tilgungszahlungen, das Abrufen von Guthaben und Darlehen und die Beendigung des Vertrages durch Kündigung, Darlehensverzicht oder vollständige Tilgung. Aus diesen Angaben lassen sich weitere Größen wie Guthaben, Anspargrad, Darlehen und andere ableiten. Diese Merkmale werden jedoch nicht als Symbole in den Sequenzen verwendet, da die Unabhängigkeit aufeinander folgender Symbole keineswegs angenommen werden kann. So gilt z. B. für die Guthabensstände eines Kontos, dass sie monoton steigend sind. Neben den oben genannten Merkmalen werden die Zugehörigkeit zur Zuteilungsphase und die Fortsetzung (die Verzögerung der Zuteilung) durch spezielle Symbole modelliert.

Zu 2.: In dieser Arbeit werden ausschließlich skalare Zeitreihen betrachtet. Dies hat den Vorteil, dass sich die Anzahl der zu bestimmenden Modellparameter gegenüber der Modellierung mit vektoriellen Sequenzen in Grenzen hält. Gleiches gilt für die Laufzeiten der verwendeten Algorithmen. Ein Nachteil kann darin liegen, dass sich relevante Strukturen in den Daten erst bei der Clusterung mehrdimensionaler Zeitreihen zeigen. In einer weiteren Ausbaustufe des Modells wäre es auf jeden Fall interessant, diesen Punkt genauer zu untersuchen. Die in dieser Arbeit vorgestellten Theorien und Algorithmen gelten uneingeschränkt auch für vektorielle Zeitreihen.

Zu 3.: Die wichtigen Aktionen Spargeldeingang und Tilgungszahlung sind in ihrer Natur kontinuierlich. Eine Diskretisierung ist immer mit gewissen Fehlern verbunden und bietet im Fall der Hidden-Markov-Modelle keine Vorteile. Das Training der Modelle kann in beiden Fällen (diskret und kontinuierlich) gleichermaßen effektiv durchgeführt werden. Insbesondere die Verwendung der Gauß-Dichte als Ausgabefunktion hat sich in vielen Anwendungen bewährt. Wenn

hingegen die kontinuierlichen Daten auf diskrete Ausgaben abgebildet werden und als Ausgabefunktion eine Ausgabematrix der Gestalt

$$b_{ij} := pr(O = j|q = i)$$

angenommen wird, so hat dies gegenüber der Gauß-Dichte den Nachteil, dass noch mehr freie Parameter trainiert werden müssen. Dies wiederum erfordert zur Gewährleistung einer vergleichbaren statistischen Sicherheit in der Parameterschätzung eine Vergrößerung der Trainingsmenge. Aus diesen Gründen werden im HMBM kontinuierliche Sequenzen eingesetzt. Die einzelnen Elemente einer Sequenz werden als Symbole bezeichnet, da sich dieser Begriff in der HMM-Literatur etabliert hat, wenngleich es sich im Fall des kontinuierlichen HMBM tatsächlich um reelle Zahlen eines kontinuierlichen Wertebereiches handelt.

Die Symbole der Sequenzen
Im Folgenden werden alle vorkommenden Aktionen und die korrespondierenden Symbole kurz beschrieben.

Spargeldeingänge: Spargeldeingänge sind prozentual zur Bausparsumme des Vertrages angegeben. Sie sind begrenzt auf das Intervall $[0, 100]$.

Kündigungen: Das Symbol der Kündigung stellt eins von drei möglichen Endsymbolen einer Sequenz dar.

Fortsetzungen: Das Fortsetzungssymbol wird für jedes Jahr, in dem sich der Vertrag im fortgesetzten Vertragszustand befindet, eingesetzt.

Zuteilungen: Das Symbol für die Zuteilung trennt in jeder Sequenz, die es betrifft, die Spargeldeingänge von den Tilgungszahlungen. Dies ist insofern bedeutsam, da für Spargeldeingänge und Tilgungen ähnliche Intervalle, aus denen die Werte stammen können, verwendet werden. Teilsequenzen, die lediglich eine Darlehensphase enthalten, wird immer das Zuteilungssymbol vorangestellt. Tritt das Zuteilungssymbol in einer Sequenz mehrfach hintereinander auf, so bedeutet dies, dass der korrespondierende Vertrag entsprechend viele Jahre im zugeteilten Vertragsszustand verharrt, ohne sich für oder gegen die Darlehensnahme zu entscheiden.

Darlehensverzichte: Das Symbol für Darlehensverzicht ist ein weiteres Symbol, mit dem eine Sequenz enden kann.

Tilgungen: Tilgungsbeiträge werden ebenso wie der Spargeldeingang prozentual zur Bausparsumme angegeben und liegen im Intervall $[0, 60]$. Der Tilgungsbeitrag eines Zeitraumes setzt sich zusammen aus der Tilgung des Darlehens und den für den Zeitraum fälligen Darlehenszinsen.

Darlehensablösungen: Dies ist der Abschluss einer Sequenz, wenn das Darlehen vollständig getilgt wurde.

Somit entstammen die möglichen Symbole für Spar- und Tilgungszahlungen einem kontinuierlichen Wertebereich, während für die übrigen Aktionen die in der Tabelle 4.1 dargestellten ganzen Zahlen als Symbole verwendet werden.

Aktion	Symbol
Kündigung	2000
Fortsetzung	3000
Zuteilung	1000
Darlehensverzicht	4000
Tilgungsende	5000

Tabelle 4.1: *Verknüpfung der Symbole mit den Sonderaktionen*

Für eine sinnvolle Modellierung muss die Zuordnung eines Symbols zu einer Aktion des Bausparers eindeutig sein. Die Unterscheidung zwischen Spargeldeingängen und Tilgungszahlungen ist dadurch gewährleistet, dass dem ersten Symbol „Tilgung“ zwingend in wenigstens einem Zeitraum das Symbol für die Zuteilung vorangestellt ist.

Der Zeitraum, aus dem die Daten stammen, erstreckt sich über die Jahre 1985 bis 1998 einschließlich. Dies hat zur Folge, dass bei allen Sequenzen, deren korrespondierender Vertag vor 1985 abgeschlossen wurde, die Anfangssymbole fehlen. Handelt es sich hierbei um Spargeldeingänge, so ist es sinnvoll und möglich, die fehlenden Symbole basierend auf weiteren vorliegenden Informationen durch folgende grobe Schätzung zu ersetzen. Die Summe der fehlenden Spargeldeingänge ergibt sich aus dem ersten vorliegenden Guthaben abzüglich der bis zu diesem Zeitpunkt gezahlten Guthabenszinsen. Im einfachsten Ansatz, der in dieser Arbeit Verwendung findet, wird diese Summe gleichmäßig auf die fehlenden Anfangsjahre aufgeteilt. Ein genaueres Vorgehen, das jedoch hier nicht implementiert wurde, bestünde darin, zur Verteilung der Summe die im ersten Zeitpunkt vorliegende Bewertungszahl zu berücksichtigen. Eine hohe BWZ wird erreicht, wenn der Schwerpunkt der Sparzahlungen im vorderen Zeitintervall liegt, während eine kleinere BWZ auf vermehrt späte Zahlungen hinweist. Somit ließe sich eine Tendenz abbilden; die exakte Verteilung der Spargeldeingänge kann jedoch auch unter Verwendung der BWZ aufgrund zu vieler unbekannter Größen nicht bestimmt werden.

Weitere assoziierte Daten
Neben den oben angegebenen Symbolen kann jede Sequenz im HMBM noch folgende weitere Merkmale enthalten:

Abschlussdatum: Das Abschlussdatum des mit der Sequenz korrespondierenden Vertrages ist wichtig für die Errechnung der Gesamtkollektivzeitreihen. Anhand des Abschlussdatums können die Sequenzen bei der Kollektivberechnung so überlagert werden, dass nur jeweils zeitgleiche Größen addiert werden. Hierfür muss aber gewährleistet sein, dass jedes Jahr durch genau ein Symbol in der Sequenz repräsentiert wird.

Bausparsumme: Jede Sequenz trägt die Bausparsumme des Vertrages in Einheiten von 1000 DM (TDM). In der Realität kommen ausschließlich Verträge vor, deren Bausparsumme ein ganzzahliges Vielfaches von 1000 DM beträgt. Wird eine Sequenz z. B. bei der Mischmodellierung auf mehrere Modelle aufgeteilt, dann ergeben sich allerdings auch nicht ganzzahlige Bausparsummen. Welche Rolle die Bausparsumme in der Modellierung spielt, wird im Abschnitt 4.6 näher erörtert.

ID: Zur Identifizierung erhält jede Sequenz eine eindeutige ID. Hierfür wird eine verschlüsselte, eindeutige Vertragsnummer verwendet.

Label: Ein optionales Label kann zur Klassifikation von Sequenzen eingesetzt werden: Jede Sequenz bekommt als Label einen Kenner des besten Modells.

4.3.2 Beschreibung der Datensätze

In dieser Arbeit werden verschiedene Datensätze, im Folgenden meist Sequenzfelder genannt, verwendet und untersucht. In der Modellierung mit dem HMBM spielen zwei unterschiedliche Arten von Sequenzfeldern eine Rolle:

Trainingssequenzen: Mit ihnen wird die Clusterung durchgeführt und sie dienen der Bestimmung sämtlicher Modellparameter.

Bestandssequenzen: Diese Sequenzen repräsentieren das aktuelle Kollektiv der Bausparkasse, dessen zeitliche Weiterentwicklung simuliert werden soll.

Für diese beiden Sequenzfelder ist es sinnvoll, eine weitere Unterteilung nach so genannten Tarifklassen vorzunehmen, da sich Verträge unterschiedlicher Tarife in ihren statistischen Verhaltensweisen in der Regel deutlich unterscheiden. Die Tarife bestimmen u. a. die Regelsparrate, den Mindestanspargrad und die Tilgungsrate. Somit ist die Tarifzugehörigkeit ein relevantes Unterscheidungsmerkmal der Sequenzen und kann als solches für eine Vorpartitionierung der Sequenzen genutzt werden. Die Einführung von Tarifklassen wurde mit der Entwicklung des mesoskopischen Simulationsmodells (siehe 4.1.2) vorgenommen. Eine Tarifklasse beinhaltet mehrere, einander ähnliche Tarife.

Alle in dieser Arbeit betrachteten Sequenzen entstammen genau einer Tarifklasse, die den Tarif 1 und den Tarif Vario 2 beinhaltet. Aus dieser Tarifklasse liegen besonders viele und lange Sequenzen vor. Die Durchführung einer Bausparsimulation in der Praxis erfordert die Berücksichtigung aller Tarife, was aber im Kontext dieser Arbeit zu keinen neuen Erkenntnissen führen würde.

Trainingssequenzen
Die Trainingssequenzen sind für die Mehrzahl der Untersuchungen aus all jenen Verträgen erzeugt worden, die im Jahr 1985 abgeschlossen wurden. Dies ist der erste Zeitraum, aus dem Daten der betrachteten Bausparkasse vorliegen, und somit weisen diese Sequenzen die größtmögliche Vollständigkeit auf. Die Mehrzahl der Sequenzen enthält die Abwicklung des korrespondierenden Vertrages, einige Sequenzen haben jedoch ein „offenes Ende“ in dem Sinne, dass der dazugehörige Vertrag bis zum Jahr 1998 sein Vertragsende nicht erreicht hat.

Die Gesamttrainingsmenge in der Tarifklasse mit Abschluss 1985 umfasst etwa 12 400 Sequenzen. Darüber hinaus werden auch kleinere Teilmengen daraus herangezogen, um den Einfluss der Größe der Trainingsmenge zu untersuchen. Auch eine Zusammenlegung mit dem Abschlussjahrgang 1986 zur Vergößerung der Trainingsmenge wird vorgenommen und deren Auswirkung auf das Trainingsergebnis wird analysiert.

Zusätzlich werden auch Untersuchungen mit Trainingsmengen durchgeführt, die sich aus zwei Gruppen von Verträgen zusammensetzen:

1. Alle Verträge eines Abschlussjahrganges, die ihre Sparphase, sei es durch Kündigung oder durch Zuteilung, beendet haben. Von diesen Sequenzen werden eventuell vorhandene Symbole der Darlehensphase entfernt.
2. Alle Verträge mit vollständiger Darlehensphase, die ihr Darlehen in einem vorgegebenen Referenzjahr vollständig getilgt haben. Aktionen der Sparphase werden bei diesen Sequenzen nicht berücksichtigt.

Mit einer solchen Trainingsmenge soll die Möglichkeit untersucht werden, verschiedene Zustände des Hidden-Markov-Modells mit unterschiedlichen Sequenzen zu trainieren. Dies ist besonders wichtig, wenn nicht genügend lange, den gesamten Vertragsablauf beschreibende Datensätze vorhanden sind. Im Abschnitt 5.2.4 wird detaillierter auf diese Problematik eingegangen.

Im Abschnitt 6.3 wird außerdem noch mit einer Trainingsmenge gearbeitet, die sich durch Stichprobenbildung aus den Gesamtmenge aller aktiven Sequenzen ergibt.

Bestandssequenzen
Ein Feld von Bestandssequenzen bezieht sich immer auf ein bestimmtes Jahr, das so genannte Bestandsjahr. Das bedeutet, es enthält alle Sequenzen und ausschließlich solche, deren korrespondierende Verträge in dem Bestandsjahr aktiv sind. Die betrachteten Mengen von Bestandssequenzen umfassen etwa 250 000 Sequenzen.

Je nach Wahl des Bestandsjahres bewirkt das zur Verfügung stehende, eingeschränkte Zeitfenster, dass der Beginn oder das Ende vieler Bestandssequenzen in den Daten nicht vorliegt. Ziel der Simulation ist es, die fehlenden Sequenzenden unter Verwendung der trainierten Hidden-Markov-Modelle zu generieren. Ein fehlender Sequenzanfang bereitet hierbei jedoch Schwierigkeiten, da beim Training (s. o. unter „Trainingssequenzen“) ausschließlich Sequenzen verwendet werden, deren Anfang in den Daten vorhanden ist. Dieses Problem wird dadurch gelöst, dass bei Sequenzen, deren Vertrag im Bestandsjahr noch in der Sparphase ist, eventuell fehlende Anfangssymbole wie unter 4.3.1 beschrieben rekonstruiert werden.

Befindet sich der Vertrag im Bestandsjahr schon in der Darlehensphase, so werden vorhandenen Sparsymbole von der Sequenz abgetrennt und die Sequenz startet mit dem Symbol für die Zuteilung.

4.4 Modelltopologie

Im Abschnitt 2.2 wurden diejenigen Elemente eines HMMs zur Modelltopologie gerechnet, die vor Beginn des Modelltrainings festgelegt werden müssen und in der Regel vom Training nicht verändert werden (eine Ausnahme wird in Abschnitt 4.4.2 beschrieben). Hier soll es nun darum gehen, die Bedeutung der Modelltopologie im Zusammenhang mit der Bausparmodellierung genauer zu erläutern.

4.4.1 Sonderzustände

Die Bausparaktionen „Kündigung“, „Fortsetzung“, „Zuteilung“, „Darlehensverzicht“ und „Tilgungsende“ werden durch eindeutige, ganze Zahlen repräsentiert (s. Abschnitt 4.3.1). Jeder dieser Aktionen ist im HMBM genau ein Zustand zugeordnet, der ausschließlich die entsprechende Zahl ausgeben kann. Dies wird dadurch erreicht, dass Zustände als unveränderbar markiert werden können. Das bedeutet, dass die einmal bei der Initialisierung gewählten Ausgabeparameter des betreffenden Zustandes vom Trainingsalgorithmus nicht angetastet werden. Im kontinuierlichen Modell werden die Ausgabefunktionen der Sonderzustände mit sehr geringer Varianz und mit einen Mittelwert, der mit der entsprechenden ganzen Zahl der Sonderaktion übereinstimmt, initialisiert. Dies führt bei der Generierung von Sequenzen dazu, dass Sonderzustände mit hinreichender Genauigkeit ausschließlich das gewünschte Symbol ausgeben können.

Antatt die Varianz mit einem sehr kleinen Wert zu initialsieren, wäre es auch möglich, für die Sonderzustände diskrete Ausgabefunktionen mit nur jeweils einem möglichen Ausgabewert zu verwenden. Dieser Weg wurde jedoch aus programmiertechnischen Gründen nicht beschritten, da er den Nachteil einer Vermischung eines diskreten mit einem kontinuierlichen HMM mit sich brächte.

Mit der Einführung der Sonderzustände ist eine eindeutige Verknüpfung zwischen einem Zustand und einem Ausgabesymbol hergestellt. Bei den Sonderzuständen handelt es sich somit nicht um versteckte, sondern um sichtbare Zustände, da ein Sondersymbol den zugrundeliegenden Zustand eindeutig identifiziert.

4.4.2 Zustandswechsel

Sehr wesentlich für den erfolgreichen Einsatz von Hidden-Markov-Modellen zur Modellierung von Bausparkollektiven ist die Möglichkeit, den Ablauf eines typischen Bausparvertrages durch die Vorgabe einer geeigneten Modelltopologie zu erfassen. Im Abschnitt 4.1.1 wurde erläutert, dass der Ablauf eines Bausparvertrages in drei Phasen (Sparphase, Zuteilungsphase und Darlehensphase) unterteilt werden kann. Diese Phasen werden in der HMM-Modellierung durch drei disjunkte Gruppen von Zuständen realisiert. Die drei Phasen werden von jedem vollständigen Vertrag (ein Vertrag, bei dem der Bausparer weder kündigt, noch auf das Darlehen verzichtet) in der Reihenfolge Sparphase, Zuteilungsphase, Darlehensphase durchlaufen. Diese Tatsache kann in der Wahl der HMM-Topologie ausgenutzt werden, indem nur solche Zustandswechsel erlaubt werden, die ein korrektes Durchlaufen der drei Phasen sicherstellen. Konkret bedeutet dies, dass die Einträge der Übergangsmatrizen, die verbotene Zustandswechsel repräsentieren, mit null initialisiert werden. Diese Einträge a_{ij} der Übergangsmatrix mit $a_{ij} = 0$ sind Teil der Modelltopologie, da sie vom Trainingsalgorithmus nicht verändert werden. Jeder Nulleintrag in der Übergangsmatrix führt folglich auch zu einer Reduktion der Anzahl der freien Parameter.

Im Zuge der Parameter-Reestimierung kann es vorkommen, dass für bestimmte Zustandswechsel eine verschwindende Wahrscheinlichkeit berechnet wird. Somit kann das Modelltraining zu einer Änderung der Modelltopologie führen. Besonders bei kleinen Trainingsmengen kann dies ein unerwünschter Effekt sein, der aber in einem Nachbearbeitungsschritt behoben werden kann,

indem die betreffenden Parameter auf einen vorzugebenden Minimalwert angehoben und die übrigen Parameter entsprechend reskaliert werden [32].

Aus theoretischer Sicht wäre es nicht zwingend notwendig, verbotene Phasenübergänge bereits bei der Modellinitialisierung auszuschließen, da der Trainingsalgorithmus eigenständig in der Lage ist, die entsprechenden Übergänge mit der Wahrscheinlichkeit null zu belegen, wenn sie durch die Trainingsdaten nicht gestützt werden. Dennoch ist es angebracht, den Trainingsalgorithmus mit einer entsprechend dünnen Übergangsmatrix zu initialisieren, da dies die beiden folgenden wesentlichen Vorteile gegenüber dem Start mit einer vollständig besetzten Übergangsmatrix bietet:

1. Die Anzahl der freien und damit zu schätzenden Parameter verringert sich. Dies erhöht bei gegebenem Umfang der Trainingsmenge die statistische Sicherheit.
2. Die Laufzeit des Trainingsalgorithmus reduziert sich, da effektiv weniger als N^2 mögliche Zustandswechsel zu berücksichtigen sind.

Der zweite Punkt wird im HMBM durch eine entsprechende Implementierung der Übergangswahrscheinlichkeiten ausgenutzt. Anstatt die vollständige Übergangsmatrix zu speichern, werden lediglich die Übergänge mit einer Wahrscheinlichkeit größer null in Form einer linearen Liste gespeichert. Diese Listen müssen zu keinem Zeitpunkt des Trainings erweitert werden, da ursprünglich verbotene Übergänge auch nach Durchführung des Trainings verboten sind [32].

Im Abschnitt 5.4.4 werden verschiedene Modelltopologien hinsichtlich ihrer Eignung zur Modellierung von Bausparkollektiven untersucht. Ein Beispiel mit zwölf Zuständen ist in Abbildung 4.3 dargestellt:

Wieder stellen die Knoten des Graphen die verschiedenen Zustände des Modells dar. Die quadratisch gezeichneten Zustände sind die oben erwähnten Sonderzustände und ein weiterer Zustand (genannt „Nullspar“), der dadurch charakterisiert ist, dass ausschließlich die Zahl Null ausgegeben wird.

Eine sinnvolle Wahl der Initialwahrscheinlichkeit π zur Modellierung vollständiger Sequenzen wäre bei dieser Modelltopologie:

$$\pi_i \begin{cases} > 0 & \text{falls} \quad 0 \leq i \leq 4 \\ = 0 & \text{falls} \quad 5 \leq i \leq 11 \end{cases}, \quad \sum_{i=0}^{4} \pi_i = 1.$$

Hierdurch wird gewährleistet, dass Zustandspfade ausschließlich mit Zuständen der Sparphase beginnen dürfen. Die Zustände null bis fünf gehören zur Sparphase, die Zustände sechs und sieben markieren die Zuteilungsphase und die restlichen Zustände gehören der Tilgungsphase an. Durch die Wahl der Übergangsmöglichkeiten, wie in Abbildung 4.3 dargestellt, ist die korrekte Abfolge der verschiedenen Phasen eines Bausparvertrages gewährleistet.

Die Knoten vier, sieben und elf besitzen keine Ausgangskanten. Das bedeutet, wenn sich das System in einem dieser Zustände befindet, bricht die Sequenz ab. Dies entspricht genau den drei Möglichkeiten, die ein Bausparer besitzt, seinen Bausparvertrag zu beenden.

Die erlaubten Übergänge der Sparphase stellen ein Beispiel dar. Andere Konstellationen, insbesondere vollständig verbundene Sparzustände, sind möglich und werden noch im Abschnitt 5.4.4 untersucht.

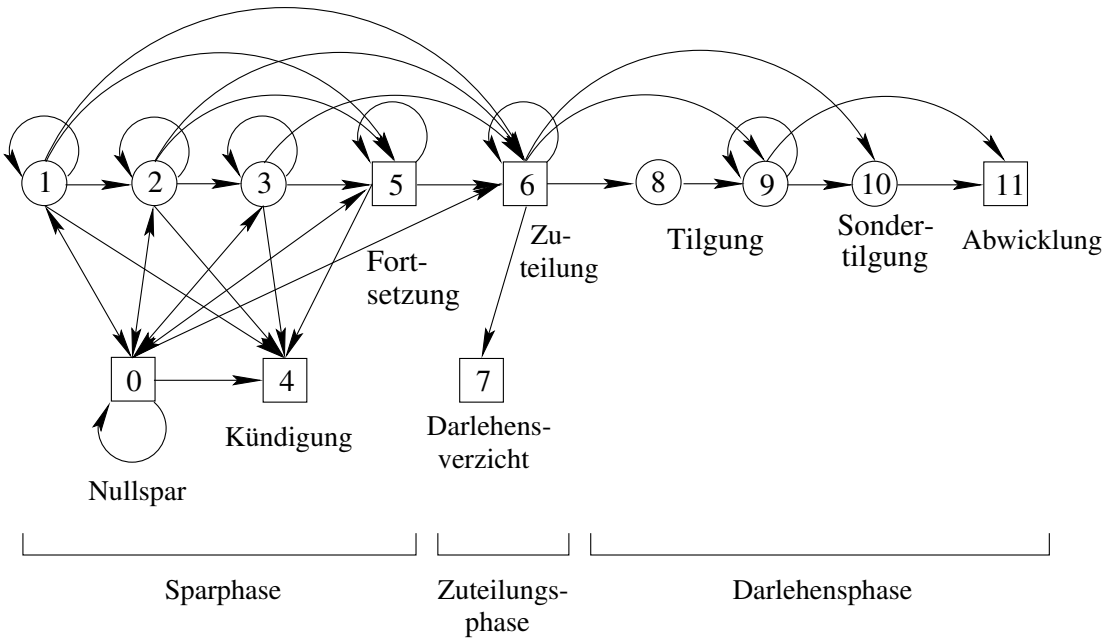

Abbildung 4.3: Eine „strikte Links Rechts“ Topologie mit Nullzustand (SLR0) zur Abbildung eines vollständigen Vertragsablaufes

Die Zustände acht und zehn der Tilgungsphase besitzen im Gegensatz zum Zustand neun keine Selbstübergänge. Das hängt damit zusammen, dass sie zwei in der Regel jeweils einmalige Aktionen widerspiegeln sollen. Ein Vertrag leistet im ersten Jahr der Darlehensphase im Mittel eine Tilgung, die geringer ist als die Regeltilgung. Dies ist ein unterjähriger Effekt; der entsprechende Vertrag ist erst im Laufe des betreffenden Jahres in die Darlehensphase eingetreten und tilgt folglich nicht den vollständigen Betrag eines Jahres. Hierfür wird der Zustand acht eingeführt. Der Zustand zehn bietet die Möglichkeit, eine einmalige Sondertilgung zur vollständigen Ablösung des Darlehens abzubilden. Dies ist eine in den realen Daten häufig auftretende Aktion, um den Vertragsablauf vorzeitig zu beenden.

4.4.3 Modelldimension

Unter der Dimension eines Hidden-Markov-Modells soll hier die Anzahl der freien, also der trainierbaren Modellparameter verstanden werden. Die Modelldimension hängt ab von der Anzahl der

- Zustände,
- Nulleinträge in A und π,
- Übergangsklassen,
- Parameter der Ausgabefunktionen,
- Ausgabefunktionen pro Zustand.

Im Zusammenhang mit der Clusterung spielt außerdem die Anzahl der Cluster bzw. die Anzahl der Komponentenmodelle eine Rolle. Somit ist die Modelldimension eine abgeleitete Größe, die sich aus der Modelltopologie ergibt.

Die Dimension des Hidden-Markov-Modells hat Einfluss auf zwei gegenläufige, elementare Anforderungen, die ein HMM möglichst gut erfüllen sollte:

Spezialisierung: Das Modell soll sich an die vorgelegten Trainingsdaten gut anpassen können, um so die Eigenschaften der Trainingsmenge gut abzubilden. Dies gelingt im Allgemeinen umso besser, je größer die Modelldimension ist. Wenn im Extremfall für jede Trainingssequenz eine exklusive Abfolge von Zuständen gewählt werden kann, so ergibt sich bei nicht nach unten beschränkter Varianz der Ausgabefunktion eine unendliche Likelihood, da sich die Ausgabe jedes Zustandes auf genau ein Symbol spezialisiert.

Generalisierung: Das Modell soll in der Lage sein, Eigenschaften der Trainingsmenge zu generalisieren. Das bedeutet, dass ähnliche, aber nicht identische Eigenschaften anderer Sequenzen erkannt werden sollen. Dies spielt bei der Sprach- und Mustererkennung und bei der Bausparmodellierung eine ganz entscheidende Rolle. Das obige Extrembeispiel verdeutlicht, dass eine zu groß gewählte Modelldimension der Fähigkeit zur Generalisierung entgegensteht.

In der Bausparmodellierung mit Hidden-Markov-Modellen sind beide oben erwähnten Anforderungen relevant. Bei der Generierung neuer Sequenzen sollen die statistischen Eigenschaften der Trainingsmenge gut reproduziert werden, wofür sich das Modell auf die Trainingsmenge spezialisieren muss. Andererseits sollen Nicht-Trainingssequenzen sinnvoll den zur Auswahl stehenden Modellen der Clusterung bzw. den verschiedenen Komponenten der Mischmodellierung zuzuordnen sein. darstellt. Im Abschnitt 5.4 werden zwei Methoden zur Lösung dieser Problematik beschrieben und Eigenschaft in der Anwendung auf das HMBM untersucht.

Zwischen der Dimension eines Modells und dem Umfang der Trainingsmenge gibt es einen wichtigen Zusammenhang. Um eine gleichbleibende statistische Sicherheit bei der Parameterschätzung zu gewährleisten, muss bei Erhöhung der Modelldimension ebenfalls die Größe der Trainingsmenge erhöht werden, damit die effektiv pro zu schätzenden Parameter zugrunde liegende Datenmenge konstant bleibt. Untersuchungen zum Umfang der Trainingsmenge finden sich im Abschnitt 5.2.4.

4.4.4 Ausgabedichten

Die Verwendung der Dichtefunktion

$$f_{\mathcal{N}}(x, \mu, \sigma^2) := \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right)$$

der eindimensionalen Normalverteilung $\mathcal{N}(\mu, \sigma^2)$ als Ausgabefunktion in der Modellierung der Bauspardaten hat zur Folge, dass das Modell auch negative Werte mit nicht verschwindender Wahrscheinlichkeit ausgibt, auch wenn in den Trainingsdaten keine negativen Symbole vorhanden sind, da die Gauß-Dichte für den gesamten Definitionsbereich von $-\infty$ bis ∞ größer als

null ist. Bei eindimensionalen Daten ist die Wahrscheinlichkeit für die Ausgabe eines negativen Symbols bei gegebenen μ, σ^2 gleich

$$pr(O < 0|\mu, \sigma^2) = \int_{-\infty}^{0} f_{\mathcal{N}}(x, \mu, \sigma^2)\, dx. \tag{4.1}$$

In den Trainingsdaten kommt der Wert Null in der Sparphase recht häufig vor. Das bedeutet, dass sich für manche Zustände ein $\mu \approx 0$ einstellen wird. Wenn dort außerdem die Varianz deutlich größer als null ist, so ergibt sich nach Gleichung 4.1 eine nicht zu vernachlässigende Wahrscheinlichkeit für die Generierung negativer Werte.

Es gibt zwei Möglichkeiten, dies zu verhindern oder zumindest in der Häufigkeit einzuschränken. Die erste besteht darin, eine entsprechende Ausgabefunktion einzusetzen, deren Wahrscheinlichkeit zur Ausgabe negativer Werte gleich null ist. In dieser Arbeit wird hierfür die bei null abgeschnittene Gauß-Dichte verwendet, die wie folgt definiert ist:

$$\bar{f}_{\mathcal{N}}(x, \mu, \sigma^2) := \begin{cases} 0 & \text{falls} \quad x < 0 \\ \left(\int_0^{\infty} f_{\mathcal{N}}(y, \mu, \sigma)\, dy\right)^{-1} f_{\mathcal{N}}(x, \mu, \sigma) & \text{falls} \quad x \geq 0 \end{cases}.$$

Mit dieser Wahrscheinlichkeitsdichte werden nur positive Symbole ausgegeben und somit ergibt sich eine bessere Abbildung der realen Bausparsequenzen. Allerdings führt die Verwendung der abgeschnittenen Gauß-Dichte zu einem aufwendigeren Reestimierungsverfahren, da die Reestimierungsformeln der einfachen Gauß-Dichte (3.10) und (3.11) durch numerische Berechnungsmethoden ersetzt werden müssen. Das genaue Verfahren wird ausführlich in [23] beschrieben.

Eine Alternative zur Verwendung der abgeschnittenen Normalverteilung bietet folgendes Vorgehen: Zum Training und Generieren wird die symmetrische Gauß-Dichte verwendet, jedoch wird ein „Nullzustand“ bei der Initialisierung vorgegeben, der den Mittelwert null und eine sehr geringe Varianz aufweist. Für diesen Zustand wird die in 2.4 erläuterte Ausgabefixierung eingesetzt. Die Idee dabei ist, dass die Existenz eines solchen Zustandes dazu führt, dass die Mittelwerte der übrigen Ausgabefunktionen sich auf deutlich positive Werte einstellen. Die zahlreich in den Daten vorhandenen Nullen werden überwiegend durch den Nullzustand repräsentiert, welcher durch die geringe Varianz keine signifikant negativen Werte beim Generieren von Sequenzen ausgibt. Eine konkrete Untersuchung dieser Fragestellung findet sich im Abschnitt 5.4.5.

Im Abschnitt 2.3.3 über den Baum-Welch-Algorithmus wurden die Reestimierungsformeln für den allgemeinen Fall einer Überlagerung verschiedener Gauß-Dichten angegeben. Im HMBM wird jedoch überwiegend nur eine Gauß-Dichte eingesetzt, da dies bereits hinreichend gute Ergebnisse liefert. Die Verwendung mehrerer Dichten würde zu einer Erhöhung der freien Parameter und damit zur Vergrößung der Modelldimension führen. Eine Ausnahme wird bei der expliziten Modellierung der Bausparsumme als Clustermerkmal gemacht. Im Abschnitt 5.2.5 werden verschiedene Überlagerungen mehrerer Gauß-Dichten zur Abbildung der Bausparsumme untersucht.

4.5 Übergangsklassen

In Abschnitt 2.4 wurde eine Erweiterung von Hidden-Markov-Modellen durch die Einführung von Übergangsklassen beschrieben. Hierdurch wird der Zustandswechsel zu einem bestimmten Zeitpunkt von der bis zu diesem Zeitpunkt ausgegebenen Teilsequenz mit beeinflusst. Der zukünftige Zustand hängt somit nicht mehr ausschließlich vom aktuellen Zustand ab, sondern auch von der Historie, die durch die Anfangssequenz gegeben ist. In diesem Abschnitt wird beschrieben, wie diese Abhängigkeit modelliert wird und warum es bei der Abbildung von Bausparsequenzen vorteilhaft ist, Übergangsklassen zu verwenden.

Hintergrund dieser Erweiterung ist die Notwendigkeit, dass die bei der Durchführung einer Simulation generierten künstlichen Sequenzen gewisse strikte Bedingungen (s.u.) einhalten müssen. Sequenzen, die diese Bedingungen verletzen, sind aus bauspartechnischer Sicht verbotene Sequenzen. Versuche, diese Bedingungen mit Hidden-Markov-Modellen ohne Übergangsklassen einzuhalten, waren wenig erfolgreich [23].

Im Folgenden sei eine Sequenz O der Länge T und ein HMM mit Modellparametern λ gegeben. Mit $O_{(t)}$ wird wieder die Teilsequenz vom Zeitpunkt 1 bis zum Zeitpunkt t bezeichnet. Die Bestimmung der Übergangsklassen geschieht durch die Funktion class($O_{(t)}$) (s. Gleichung (2.39)), die für jedes $t, 1 \leq t \leq T$, die Anfangssequenz $O_{(t)}$ auf eine von L möglichen Klassen abbildet. Die Funktion class($O_{(t)}$) wird abschnittsweise definiert, abhängig davon, ob es sich bei O_t um ein Symbol der Sparphase, der Tilgungsphase oder um ein Sondersymbol wie „Fortsetzung“ oder „Zuteilung“ handelt.

Das Problem besteht darin, eine Funktion class($O_{(t)}$) so zu definieren, dass generierte Sequenzen die folgenden Anforderungen möglichst gut erfüllen:

1. Das Symbol „Zuteilung“ taucht frühestens auf, wenn die durch sämtliche Vorgängersymbole definierte Sparphase eine Zuteilung erlaubt. Das bedeutet, dass die in Abschnitt 4.1.1 erläuterten Zuteilungsbedingungen Mindestspardauer, Mindestanspargrad und Mindestbewertungszahl erfüllt sein müssen.
2. Das Ende der Tilgungsphase, markiert durch das Symbol „Darlehensende“, korrespondiert mit der vollständigen Tilgung des Darlehens.
3. Die Verteilung der Aufenthaltsdauer der sichtbaren Zustände “Fortsetzung“ und „Zuteilung“ (siehe Abschnitt 4.4) sollte von den generierten Sequenzen möglichst gut approximiert werden.

Die ersten beiden Bedingungen werden von allen realen Sequenzen erfüllt, da sie strikte bauspartechnische Nebenbedingungen darstellen, die von der Bausparkasse gewährleistet werden. Bei der Generierung von künstlichen Sequenzen mit einem HMM ist dies jedoch nicht automatisch sichergestellt, doch bietet die Einführung von Übergangsklassen eine Lösung dieser Problematik.

Einhaltung von Zuteilungsbedingungen
Zur Berechnung von class($O_{(t)}$) in der Sparphase ist es zunächst notwendig, eine Bewertungszahl bwz($O_{(t)}$) zu definieren. In Übereinstimmung mit der Berechnungsmethode der untersuchten

Bausparkasse wird die Bewertungszahl einer Sequenz unter Vorgabe dreier Konstanten α, β, γ wie folgt definiert:

$$\text{bwz}(O_{(t)}) := \frac{1}{\gamma}(\alpha\,\text{guth}(O_{(t)}) + \beta\,\text{szins}(O_{(t)})). \tag{4.2}$$

Dabei kennzeichnet $\text{guth}(O_{(t)})$ das durch die Teilsequenz $O_{(t)}$ resultierende relative Guthaben einschließlich Zinsen zum Ablauf des Jahres *t*. Diese Größe wird auch als Anspargrad bezeichnet. Mit $\text{szins}(O_{(t)})$ werden die angesammelten relativen Sparzinsen bezeichnet. Die üblicherweise bei der Berechnung der Bewertungszahl eingehende Bausparsumme muss nicht explizit berücksichtigt werden, da sie bereits in den beiden relativen Größen Guthaben und Zinsen enthalten ist.

Da die Spargeldeingänge in der Sequenz jahresweise vorliegen, also unterjährige Verteilungen der Spargeldeingänge verdeckt bleiben, müssen die Zinsen und folglich auch das Guthaben näherungsweise berechnet werden. Die nachfolgend angegebenen rekursiven Formeln zur Approximation von Guthaben und Zinsen, in denen *Z* den Guthabenszinssatz und T_z den Zeitpunkt der Zuteilung bezeichnen, gelten zunächst nur für solche Sequenzen, die in der Sparphase kein Sondersymbol „Fortsetzung“ enthalten. Sie lassen sich jedoch einfach dadurch erweitern, dass für die Zeitpunkte *t*, zu denen einen Fortsetzung vorliegt, O_t auf null gesetzt wird.

Für $t = 1$:

$$\begin{aligned} \text{szins}(O_{(1)}) &:= \frac{Z}{2}O_1, \\ \text{guth}(O_{(1)}) &:= (1+\frac{Z}{2})O_1. \end{aligned}$$

Für $t = 2, \ldots, T_z - 1$:

$$\begin{aligned} \text{szins}(O_{(t)}) &:= (1+Z)\,\text{szins}(O_{(t-1)}) + Z\,\text{guth}(O_{(t-1)}) + \frac{Z}{2}O_t, \\ \text{guth}(O_{(t)}) &:= \sum_{\tau=1}^{t} O_\tau + \text{szins}(O_{(t)}). \end{aligned}$$

Die Definition der Übergangsklassenfunktion für eine Teilsequenz $O_{(t)}$ in der Sparphase lautet folgendermaßen:

$$\text{class}_S(O_{(t)}) := \min\left(\text{floor}\left(\frac{L\cdot\text{bwz}(O_{(t)})}{\text{Mindest-BWZ}}\right), L\right). \tag{4.3}$$

Die Funktion floor$(\cdot)$ rundet auf die nächst kleinere ganze Zahl ab und „Mindest-BWZ“ ist die tarifabhängige Mindest-Bewertungszahl, die ein Vertrag für die Zuteilung erreicht haben muss. Falls $\text{class}_S(O_{(t)}) = L$ gilt, dann wird in einem zweiten Schritt noch überprüft, ob die Sequenz auch die Mindestspardauer und den Mindestanspargrad erreicht hat. Falls dies nicht der Fall ist, wird $\text{class}_S(O_{(t)})$ auf $L - 1$ gesetzt. Auf diese Art und Weise wird erreicht, dass eine Sequenz

genau dann die größte Übergangsklasse L zugewiesen bekommt, wenn der dazugehörige Bausparvertrag alle Zuteilungsbedingungen erfüllt hat:

$$\text{class}_S(O_{(t)}) = L \;\leftrightarrow\; O_{(t)} \text{ erfüllt alle Zuteilungsbedingungen.} \tag{4.4}$$

Vorausgesetzt in den Trainingsdaten geht die Zuteilung in jedem Fall einher mit der Erfüllung aller Zuteilungsbedingungen [1], dann ist durch (4.4) gewährleistet, dass nach dem Training eines HMMs ausschließlich in der Übergangsklasse L nicht verschwindende Übergangswahrscheinlichkeiten in den Zustand, der die Zuteilung repräsentiert, vorliegen. In allen anderen Übergangsklassen ist kein Übergang in den Zuteilungszustand möglich. Wird ein solches Modell zum Generieren künstlicher Sequenzen verwendet, so ist sichergestellt, dass alle generierten Sequenzen die erste der oben aufgeführten Nebenbedingungen erfüllen.

Korrektes Tilgungsende

Für die folgenden Überlegungen wird die Zeitachse so verschoben, dass O_1 die erste Tilgungszahlung repräsentiert, da sich hierdurch einige Formulierungen vereinfachen. $O_{(t)}$ kennzeichnet weiterhin die Anfangsteilsequenz bis zum Zeitpunkt t, wobei Symbolen der Sparphase entsprechend negative Zeitpunkte zugeordnet werden. Bei der Generierung künstlicher Sequenzen muss sichergestellt sein, dass mit Erreichen des Sequenzendes die vollständige Tilgung des Darlehens erfolgt ist. Das bedeutet, dass die Summe der Tilgungen dem Anfangsdarlehen entsprechen sollte.

Mit $\text{darl}(O_{(t)})$ wird im Folgenden der mit der Sequenz O korrespondierende Darlehensstand zum Ende des Jahres t bezeichnet und $\text{tilg}(O_{(t)})$ stellt die reine Tilgung im Zeitraum t dar. Das Darlehen zum Zeitpunkt $t = 0$ wird als Anfangsdarlehen definiert (ohne Gebühren):

$$\text{darl}(O_{(0)}) := 100 - \text{guth}(O_{(0)}). \tag{4.5}$$

Für $t = 1, \ldots, T$ ergibt sich dann das Darlehen durch

$$\text{darl}(O_{(t)}) := \text{darl}(O_{(t-1)}) - \text{tilg}(O_{(t)}). \tag{4.6}$$

Die Tilgung $\text{tilg}(O_{(t)})$ wiederum ist definiert durch

$$\text{tilg}(O_{(t)}) := O_t - \text{dzins}(O_{(t)}), \tag{4.7}$$

wobei $\text{dzins}(O_{(t)})$ die im Zeitraum t zu zahlenden Darlehenszinsen sind. Erneut ergibt sich das Problem, dass die Zinsen aufgrund der Unkenntnis der unterjährigen Zahlungen nur approximiert werden könnten. Das geschieht mittels folgender Gleichung:

$$\text{dzins}(O_{(t)}) := DZ \cdot (\text{darl}(O_{(t-1)}) - \frac{\text{tilg}(O_{(t)})}{2}), \quad DZ = \text{Darlehenszinssatz.} \tag{4.8}$$

Gleichung (4.8) in Gleichung (4.7) eingesetzt ergibt somit folgende Näherung der Tilgung im Zeitraum t:

$$\text{tilg}(O_{(t)}) = \frac{O_t - DZ\,\text{darl}(O_{(t-1)})}{1 - \frac{DZ}{2}}. \tag{4.9}$$

[1] Vereinzelt können Verträge aufgrund einer ungenauen Approximation der Zinsen oder nicht erfasster Spargeldeingänge hiervon abweichen.

Über die Gleichungen (4.5), (4.6) und (4.9) kann jeder Sequenz in der Darlehensphase zu jedem Zeitpunkt ein Darlehensstand zugeordnet werden, der von der Funktion $\text{class}_D(\cdot)$ wie folgt zur Berechnung der Übergangsklassen in der Darlehensphase eingesetzt wird:

$$\text{class}_D(O_{(t)}) := \text{floor}\left((L-1)\min\left(1-\frac{\text{darl}(O_{(t)})-UD}{AD},1\right)\right)+1. \tag{4.10}$$

Die Konstante *AD* bezeichnet das aus statistischen Untersuchungen zu ermittelnde mittlere Anfangsdarlehen der Verträge. Dieses Anfangsdarlehen liegt in der Regel um einen gewissen Betrag unter dem maximal erzielbaren Anfangsdarlehen von 50% bzw. 60% (je nach Tarif), da viele Verträge zum Zeitpunkt der Zuteilung einen Anspargrad erreicht haben, der den Mindestanspargrad übersteigt. Dies führt zu einer Verringerung des Startdarlehens. Die Konstante *UD* kennzeichnet den unteren Darlehensstand. Das bedeutet, dass ein Vertrag erst dann in die höchste Übergangsklasse eintritt, wenn das Restdarlehen unterhalb der Grenze *UD* gefallen ist.

Die durch Gleichung (4.10) definierte Klassenfunktion für die Tilgungsphase bewirkt, dass vor der ersten Tilgung die Klasse 1 eingenommen wird und dass außerdem gilt:

$$\text{class}_D(O_{(t)}) = L \leftrightarrow \text{darl}(O_{(t)}) \leq UD.$$

Mit der Wahl der Übergangsklassen gemäß Gleichung (4.10) wird erreicht, dass nach dem Training der Modelle Übergänge von einem Tilgungszustand in den Endzustand der Tilgungsphase in den hohen Übergangsklassen mit großer Wahrscheinlichkeit stattfinden, während in den niedrigen Klassen nur eine sehr geringe Wahrscheinlichkeit für ein solches Ereignis vorliegt. Das ist genau der gewünschte Effekt, denn beim Generieren künstlicher Sequenzen führt dies dazu, dass die Wahrscheinlichkeit für das Erreichen des Tilgungsendes bei hohem Restdarlehen gering ist, während sie bei kleinem Restdarlehen groß ist.

Fortsetzung und Zuteilung

Neben der Einhaltung von Zuteilungsvoraussetzungen und der korrekten Abwicklung der Darlehensphase dienen die Übergangsklassen auch der Anpassung der Verteilung der Aufenthaltsdauer in speziellen Zuständen an die Daten. In dem HMBM sind dies die Zustände „Fortsetzung“ und „Zuteilung“.

In einem herkömmlichen Hidden-Markov-Modell ohne Übergangsklassen ergibt sich für die Wahrscheinlichkeit, in einem Zustand *i* genau *t* aufeinander folgende Zeitschritte zu bleiben, wenn vorrausgesetzt wird, dass sich das Modell im ersten Zeitschritt in *i* befindet, durch:

$$pr(q_2 = \ldots = q_t = i, q_{t+1} \neq i | \lambda, q_1 = i) = a_{ii}^{t-1}(1-a_{ii}). \tag{4.11}$$

Da die Übergangsmatrix *A* nicht von der Zeit abhängt, ist die Wahrscheinlichkeit, einen gegebenen Zustand zu verlassen, unabhängig davon, wie lange sich das System bereits in dem Zustand befindet. Dies führt zu einem exponentiellen Abfall der Aufenthaltswahrscheinlichkeit mit der Zeit. In vielen Anwendungen ist diese Modellierungseigenschaft störend [32] und die Einführung von Übergangsklassen kann zu einer angemesseneren Beschreibung der Daten führen.

Am Beispiel des Zustandes „Zuteilung“ soll demonstriert werden, wie die Bestimmung der Übergangsklassen im HMBM definiert ist. Für den Zustand „Fortsetzung“ gelten exakt die gleichen Aussagen. Zur Vereinfachung der Schreibweise wird angenommen, dass zum Zeitpunkt $t = 1$ zum ersten Mal das Symbol „Zuteilung“ vorliegt. Aufgrund der Topologie des HMBM, wie sie im Abschnitt 4.4 vorgestellt wurde, kann das Modell nicht in den Zustand „Zuteilung“ zurückkehren, wenn es ihn einmal verlassen hat. Analog dazu kann eine Folge von Zuteilungssymbolen in einer Sequenz niemals durch Nicht-Zuteilungssymbole unterbrochen werden. Wenn zum Zeitpunkt t noch das Symbol „Zuteilung“ vorliegt, dann wird die Funktion zur Übergangsklassenbestimmung wie folgt definiert:

$$\text{class}_Z(O_{(t)}) := \min(t, L). \tag{4.12}$$

Der Definition von $\text{class}_Z(O_{(t)})$ ist wesentlich einfacher als die von $\text{class}_S(O_{(t)})$ und $\text{class}_D(O_{(t)})$, da

1. keine „harten“ Bausparbedingungen (wie Mindestanspargrad in der Sparphase und vollständige Darlehenstilgung in der Darlehensphase) einzuhalten sind und
2. die Annahme gemacht wird, dass die Aufenthaltsdauer in der Zuteilungsphase unabhängig von den Sparzahlungen ist. Bei einer Verfeinerung des Modells wäre es eventuell lohnenswert, hier auch Untersuchungen mit komplizierteren Klassenfunktionen durchzuführen, die die Spar- und Zuteilungsphase miteinander verknüpfen.

Die Definition nach Gleichung (4.12) impliziert, dass die Zustandswechsel in der Zuteilungsphase im Wesentlichen davon abängen, wie weit die Zuteilung zurückliegt. Mit jedem Zeitschritt wird die Übergangsklasse gewechselt und die aus dem Training resultierenden Übergangsmatrizen können sich optimal an die Gegebenheiten in den Daten anpassen.

Zeitinhomogene Markov-Ketten und Markov-Ketten höherer Ordnung
Die Definition der Übergangsklassenfunktion durch Gleichung (4.12) für den Zustand Zuteilung zeigt, dass mit dieser Erweiterung der Zustandswechsel durch eine zeitinhomogene Markov-Kette modelliert werden kann. Die zeitunabhängigen Parameter a_{ij} im konventionellen HMM werden durch zeitabhängige $a_{ij}(t)$ ersetzt mit diskreten Zeitschritten t.

Somit ist das Konzept der Übergangsklassen eine sehr allgemeine Erweiterung, da es sowohl zeitinhomogene Markovketten abbilden kann, als auch in der Lage ist, die Zustandswechsel mit vergangenen Ausgabesymbolen zu verknüpfen (Gleichungen (4.3), (4.10)). Außerdem ist es prinzipiell möglich, auch wenn davon im HMBM kein Gebrauch gemacht wird, die Wahl der Übergangsklasse von mehreren Vorgängerzuständen abhängen zu lassen. Damit ergeben sich Parallelen zu Markov-Ketten höherer Ordnung. Diese reichhaltige Erweiterung des Modells ist möglich ohne Verzicht auf die Anwendbarkeit der effektiven HMM-Basisalgorithmen. Die Algorithmen können erweitert werden unter Beibehaltung der grundsätzlichen Struktur und die Ordnung der benötigten Laufzeit bleibt unverändert [23].

4.6 Modellierung der Bausparsumme

In Abschnitt 4.3.1 wurde bereits erwähnt, dass jede Sequenz die Bausparsumme des korrespondierenden Bausparvertrages in Einheiten von 1000 DM trägt. Da die Sequenz die relativen Spar-

geldeingänge enthält, wird die Bausparsumme benötigt, um die relativen Zahlungen in absolute Größen umzurechnen. In einer Kollektivsimulation sind letztlich die Geldbeträge (Bausparsumme, Guthaben, Einzahlung usw.) relevant, die Stückzahl von Verträgen bzw. Sequenzen spielt dagegen nur eine untergeordnete Rolle.

Die Bausparsumme kann auch als Gewichtung der Sequenz angesehen werden. Dies hat große Relevanz beim Training der Hidden-Markov-Modelle. Zur Veranschaulichung wird zunächst dargestellt, welche Rolle die Bausparsumme bei der Clusterung mit dem K-Means-Verfahren im mesoskopischen Modell (4.1.2) spielt. Dort ist der Zentralpunkt eines Clusters der Schwerpunkt der in dem Cluster liegenden, mit der Bausparsumme gewichteten Datenpunkte. Wenn mit x_i die Datensätze, mit x_{ij} das Merkmal j vom Datensatz i und mit w_i das dazugehörige Gewicht bezeichnet werden, dann ergibt sich das Merkmal j des Zentralpunktes Z_j folgendermaßen:

$$Z_j := \frac{\sum\limits_{i \text{ mit } x_i \in Z} w_i x_{ij}}{\sum\limits_{i \text{ mit } x_i \in Z} w_i}, \tag{4.13}$$

wobei jeweils über alle Datenpunkte summiert wird, die dem Zentralpunkt zuzurechnen sind. Jeder Datenpunkt hat proportional zu seiner Gewichtung Einfluss auf die Lage des Schwerpunktes. In Gleichung (4.13) würde sich genau der gleiche Schwerpunkt ergeben, wenn jedes Datum x_i mit Gewicht w_i durch w_i identische Datenpunkte mit Einheitsgewichtung ersetzt würde.

Übersetzt in das Vokabular der Hidden-Markov-Modelle bedeutet dies, dass jede Sequenz die Parameter des trainierten Modells proportional zu ihrer Gewichtung mitbestimmen soll. Sequenzen mit großer Bausparsumme sollten viel, Sequenzen mit kleiner Bausparsumme wenig Einfluss auf die trainierten Modellparameter haben. Dies läß sich gerade durch die oben erwähnte Vervielfältigung der Sequenzen entsprechend ihrer Gewichtung erreichen. Ausgehend davon, dass es äquivalent ist, mit gewichteten Sequenzen oder mit entsprechend vervielfältigten Sequenzen zu arbeiten, wird im Folgenden am Beispiel der Initialparameter π gezeigt, wie sich die Baum-Welch-Reestimierungsformeln auf den Fall gewichteter Sequenzen erweitern lassen, indem die Formeln für vervielfältigte Sequenzen hergeleitet werden.

Gewichtetes HMM-Training
Gegeben ist eine Menge von Sequenzen $O = (O^1, \ldots, O^K)$ mit den ganzzahligen positiven Gewichten $w = (w^1, \ldots, w^K)$. Die Menge von Sequenzen $\tilde{O}$ ergibt sich aus O dadurch, dass jedes $O^k \in O$ genau w^k-fach vervielfältigt wird:

$$\tilde{O} = (\underbrace{O^1, O^1, \ldots, O^1}_{w^1 \text{mal}}, \ldots, \underbrace{O^K, \ldots, O^K}_{w^K \text{mal}}).$$

Wie bisher, wird auch hier angenommen, dass alle betrachteten Sequenzen voneinander unabhängig sind.

Damit ergibt sich die Log-Likelihood-Funktion bei gegebenem $\tilde{O}$ wie folgt:

$$\begin{aligned} \log(L_{\tilde{O}}(\lambda)) &= \log(p(\tilde{O}|\lambda)) \\ &= \log\left(\prod_{k=1}^{K} p(O^k|\lambda)^{w^k}\right) \\ &= \sum_{k=1}^{K} w^k \log(p(O^k|\lambda)). \end{aligned} \tag{4.14}$$

In Anlehnung an Gleichung (2.52) läßt sich für diese Log-Likelihood-Funktion folgende Q-Funktion berechnen:

$$\begin{aligned} Q(\lambda, \lambda^i) &= E_q\left[\log(p(\tilde{O}, \tilde{Q}|\lambda)) \,\middle|\, \tilde{O}, \lambda^i\right] \\ &= \sum_{k=1}^{K} w^k \sum_{q \in \mathcal{Q}_{T^k}} \log(pr(O^k, q|\lambda))\, pr(q|O^k, \lambda^i). \end{aligned} \tag{4.15}$$

Die innere Summe der zweiten Gleichung erstreckt sich über alle möglichen Zustandspfade q der Länge T^k. Diese Q-Funktion kann in drei unabhängig zu maximierende Teile aufgespalten werden und mit den gleichen Überlegungen, die zu Gleichung (2.54) geführt haben, ergibt sich folgende Q-Funktion für den Parameter π:

$$Q_\pi(\lambda, \lambda^i) = \sum_{j=1}^{N} \log(\pi_j) \sum_{k=1}^{K} w^k\, p(q_1 = j, O^k|\lambda^i). \tag{4.16}$$

Diese Funktion gilt es bezüglich der π_j zu maximieren, unter der Nebenbedingung $\sum_{j=1}^{N} \pi_j = 1$. Der Einsatz der Lagrange-Parameter- Methode führt für jedes $l, 1 \leq l \leq N$ auf die folgende Gleichung:

$$\frac{\partial}{\partial \pi_l}\left[\sum_{j=1}^{N}\left(\log(\pi_j)\sum_{k=1}^{K} w^k\, p(q_1 = j, O^k|\lambda^i)\right) + \gamma\left(\sum_{j=1}^{N}\pi_j - 1\right)\right] = 0.$$

Die Ableitung berechnet ergibt:

$$\pi_l = -\frac{1}{\gamma}\sum_{k=1}^{K} w^k\, p(q_1 = l, O^k|\lambda^i). \tag{4.17}$$

Wird diese Gleichung über l summiert und die Nebenbedingung für π eingesetzt, so ergibt sich für γ:

$$\gamma = -\sum_{j=1}^{N}\sum_{k=1}^{K} w^k\, p(q_1 = j, O^k|\lambda^i).$$

Dies in Gleichung (4.17) eingesetzt ergibt schließlich die gesuchte Reestimierungsformel für π_l:

$$\pi_l = \frac{\sum_{k=1}^{K} w^k \, p(q_1 = l, O^k|\lambda^i)}{\sum_{j=1}^{N}\sum_{k=1}^{K} w^k \, p(q_1 = j, O^k|\lambda^i)}. \tag{4.18}$$

Die Erweiterung auf gewichtete Sequenzen für die übrigen Reestimierungsformeln kann auf analoge Art und Weise durchgeführt werden.

Durch Gleichung (4.18) und die entsprechenden Formeln für die übrigen Modellparameter wird die Q-Funktion aus Gleichung (4.15) niemals verschlechtert und das korrespondierende Baum-Welch-Training liefert ein lokales Optimum der Log-Likelihood-Funktion gemäß Gleichung (4.14). Die Herleitung der Reestimierungsformel (4.18) gilt auch für nichtganzzahlige w^k. Daher kann die Parameterreestimierung so auch für beliebige positive Sequenzgewichtungen eingesetzt werden. Dies ist wichtig bei der gewichteten Version des MIX-HMM-Algorithmus.

Gewichtete HMM-Clusterung
Welche Rolle spielt die Bausparsumme bei der Clusterung von Sequenzen? Zunächst sei der Fall der partitionierenden Clusterung, wie sie in Abschnitt 2.3.5 beschrieben wurde, betrachtet. Die Gewichtung hat keinen Einfluss auf die Klassifikation der Sequenzen (s. Gleichung (2.35)), da die für die Klassifikation relevanten Größen $p(O^k|\lambda_i)$ von ihr unabhängig sind. Jedoch muss das Baum-Welch-Training durch die oben beschriebene gewichtete Version ersetzt werden. Außerdem ist es notwendig, die Zielfunktion anzupassen. In Analogie zur Gleichung (4.14) muss der Beitrag jeder Sequenz zur Zielfunktion mit der jeweiligen Bausparsumme gewichtet werden, sodass die gewichtete Version der Gleichung (2.37) unter Zugrundelegung der dort gemachten Vorgaben die folgende Gestalt annimmt:

$$Z_{MD}(O, \Lambda) = \sum_{i=1}^{C} \sum_{k, O^k \in \mathcal{C}_i} w^k \log(p(O^k|\lambda_i)). \tag{4.19}$$

Das gewichtete Baum-Welch-Training bewirkt mit jeder Iteration, dass die Log-Likelihood-Funktion (4.14) verbessert wird oder konstant bleibt und somit wird auch die Zielfunktion (4.19) durch die gewichtete HMM-Clusterung mit keiner Iteration verschlechtert und das Verfahren konvergiert, wenn $Z_{MD}(O, \Lambda)$ nach oben beschränkt ist.

Gewichteter MIX-HMM-Algorithmus
Die gewichtete Zielfunktion im MIX-HMM-Algorithmus ergibt sich mit den gleichen Überlegungen wie oben und mit der Gleichung (3.12) zu:

$$Z_{MIX}(O, \Lambda, \alpha) = \sum_{k=1}^{K} w^k \log\left(\sum_{j=1}^{C} \alpha_j p(O^k|\lambda_j)\right). \tag{4.20}$$

Dies wiederum führt in der i-ten Iteration auf folgende gewichtete Q-Funktion:

$$\begin{aligned} Q((\alpha, \Lambda), (\alpha^i, \Lambda^i)) \;=\; & \sum_{k=1}^{K} w^k \sum_{j=1}^{C} \log(\alpha_j)\, pr(j|\Lambda^i, \alpha^i, O^k) + \\ & \sum_{k=1}^{K} w^k \sum_{j=1}^{C} \log(p(O^k|\lambda_j))\, pr(j|\Lambda^i, \alpha^i, O^k). \end{aligned} \tag{4.21}$$

Für diese Q-Funktion gilt das folgende Lemma:

Lemma 4.6.1 *Für die Q-Funktion aus Gleichung* (4.21) *gilt für ein beliebiges* (α^i, λ^i) *(unter Wahrung der üblichen Normierungsbedingungen)*

$$Q((\alpha^{i+1}, \Lambda^{i+1}), (\alpha^i, \Lambda^i)) \geq Q((\alpha^i, \Lambda^i), (\alpha^i, \Lambda^i)),$$

wenn
1.

$$\alpha_j^{i+1} = \frac{\sum_{k=1}^{K} w^k pr(j|\Lambda^i, \alpha^i, O^k)}{\sum_{k=1}^{K} w^k}, \quad 1 \leq j \leq C$$

und
2. die Parameter Λ^{i+1} *mit dem gewichteten Baum-Welch-Training, wie in diesem Abschnitt beschrieben, unter Verwendung der modifizierten Gewichte*

$$w_j^k := w^k pr(j|\Lambda^i, \alpha^i, O^k) \tag{4.22}$$

und den Initialparametern Λ^i *festgelegt werden.*

Beweis
Der erste Term der Q-Funktion

$$Q'(\alpha, \alpha^i, \Lambda^i) := \sum_{k=1}^{K} w^k \sum_{j=1}^{C} \log(\alpha_j)\, pr(j|\Lambda^i, \alpha^i, O^k)$$

hat exakt die gleiche Form wie die Q-Funktion in Gleichung (4.16) und die Nebenbedingungen sind analog. Wird π_j durch α_j und $p(q_1 = j, O^k|\lambda^i)$ durch $pr(j|\lambda^i, \alpha^i, O^k)$ ersetzt, so ergibt sich mit Gleichung (4.18), dass $Q'(\alpha, \alpha^i, \lambda^i)$ durch

$$\begin{aligned} \alpha_j^{i+1} &= \frac{\sum_{k=1}^{K} w^k pr(j|\lambda^i, \alpha^i, O^k)}{\sum_{j=1}^{C}\sum_{k=1}^{K} w^k pr(j|\lambda^i, \alpha^i, O^k)} \\ &= \frac{\sum_{k=1}^{K} w^k pr(j|\lambda^i, \alpha^i, O^k)}{\sum_{k=1}^{K} w^k} \end{aligned}$$

unter Einhaltung der Nebenbedingung $\sum_{j=1}^{C} \alpha_j = 1$ maximiert wird.

Ferner sei w_j^k für $1 \leq k \leq K$ und $1 \leq j \leq C$ gemäß Gleichung (4.22) definiert. Außerdem seien $\lambda_j^{i+1}, 1 \leq j \leq C$, die Parameter, die sich aus dem gewichteten Baum-Welch-Training mit den Gewichten w_j^k ausgehend von λ_j^i ergeben. Dann gilt

$$\sum_{j=1}^{C}\sum_{k=1}^{K} w_j^k \log(p(O^k|\lambda_j^{i+1})) \geq \sum_{j=1}^{C}\sum_{k=1}^{K} w_j^k \log(p(O^k|\lambda_j^i))$$

aufgrund der Eigenschaft des Baum-Welch-Algorithmus , die gewichtete Log-Likelihood jedes Komponentenmodells zu verbessern oder zumindest nicht zu verschlechtern. Somit werden beiden Teile der Q-Funktion (4.21) nicht verschlechtert und die Aussage des Lemmas ist bewiesen. □

Mit der Bemerkung 2.5.2 folgt aus obigem Lemma:

$$Z_{MIX}(O, \lambda^{i+1}, \alpha^{i+1}) \geq Z_{MIX}(O, \lambda^{i}, \alpha^{i}).$$

Ist die Zielfunktion $Z_{MIX}(O, \lambda, \alpha)$ nach oben beschränkt und ist durch die Initialisierung sichergestellt, dass $Z_{MIX}(O, \lambda^0, \alpha^0) > -\infty$ gilt, so ist gewährleistet, dass der Algorithmus MIX-HMM auch in der gewichteten Version konvergiert.

Bausparsumme als Clustermerkmal
Neben der Berücksichtigung der Bausparsumme in Form einer Sequenzgewichtung beim Training der Hidden-Markov-Modelle , stellt sich die Frage, ob es sinnvoll ist, die Bausparsumme auch als direktes Merkmal in die Clusterung der Sequenzen einfließen zu lassen. Im Wesentlichen sprechen zwei Punkte für ein solches Vorgehen:

1. Verträge unterschiedlicher Bausparsumme weisen tendenziell unterschiedliche Verhaltensmuster auf und somit stellt die Bausparsumme ein relevantes Clustermerkmal dar.
2. Die ermittelten Parameter der Verteilung der Bausparsumme können beim Generieren neuer Sequenzen genutzt werden, um jede Sequenz mit einer Bausparsumme zu versehen, um somit eine realitätsnähere Simulation zu ermöglichen.

Am einfachsten läßt sich die Bausparsumme als explizites Clustermerkmal mit der im Abschnitt 3.3 beschriebenen Methode der Topologieerweiterung einsetzen. Dazu wird ein zusätzlicher Zustand (im Folgenden q_{BS} genannt) zur Abbildung der Bausparsumme eingeführt und jeder Sequenz O^k wird die Bausparsumme in Form eines Gewichtes w_k vorangestellt:

$$D^k = (w^k, O^k).$$

Wenn bei der Modellinitialisierung vorgegeben wird, dass jedes Modell mit Wahrscheinlichkeit eins in q_{BS} starten muss, so führt das Training dieses Modells mit den oben definierten Sequenzen D^k dazu, dass die Ausgabefunktion von q_{BS} genau die Verteilung der Bausparsumme der Trainingsequenzen wiedergibt. Diese Verteilung könnte bei der Generierung des Neugeschäftes im HMBM genutzt werden, wenn es für die Untersuchung bestimmter Fragestellungen zweckmäßig ist, Sequenz- bzw. Vertragsanzahlen und deren Verteilungen zu betrachten. Im Abschnitt 5.2.5 werden Clusterungen unter Einbeziehung der Bausparsumme als explizites Clustermerkmal mit den herkömmlichen Clusterungen verglichen.

In diesem Kapitel wurden die wichtigsten Aspekte des Bausparens und alle Elemente des HMBM erläutert. In den beiden nachfolgenden Kapiteln geht es darum, Varianten des Modelltrainings und der Simulation mit realen Daten zu analysieren und die Eignung des Modells zur Simulation von Bausparkollektiven zu demonstrieren.

Kapitel 5

Training des HMBM

Ein geeignetes Training der Hidden-Markov-Modelle, also die datengestützte Optimierung der Modellparameter, ist die entscheidende Voraussetzung für einen sinnvollen Einsatz von Hidden-Markov-Modellen in der Simulation und Modellierung von Bausparkollektiven. Daher ist diesem Thema ein eigenständiges Kapitel gewidmet, dessen Inhalte nun skizziert werden.

Im ersten Abschnitt geht es um die Bewertung des Modelltrainings. Das ist notwendig, um verschiedene Alternativen im Training gegeneinander abzuwägen. Daran anschließend werden diese Alternativen im Training und in der Clusterung unter Verwendung realer Bauspardaten genauer untersucht. Dies umfasst u.a. verschiedene Analysen zum Training eines einzelnen Modells, den Vergleich der beiden vorgestellten Clusteralgorithmen und dabei mögliche Initialisierungen und den Einsatz verschiedener Trainingsmengen. Im dritten Abschnitt werden die Problematik der lokalen Extrema beschrieben und drei Lösungansätze zu deren Überwindung vorgestellt und miteinander verglichen. Im darauf folgenden Teil werden Aspekte der datenbasierten Modellwahl untersucht. Dies betrifft die wichtigen Fragen nach einer geeigneten Modelldimension und Modelltopologie.

5.1 Bewertung des Trainings

In den vorangegangenen Kapiteln wurde unter anderem deutlich, dass das Training eines Hidden-Markov-Modells durch sehr viele Faktoren beeinflusst wird. Als wichtigste sind hier zu nennen:

- Umfang und Zusammensetzung der Trainingsmenge
- Wahl des Clusterverfahrens (HMM-Clusterung oder MIX-HMM)
- Modelltopologie bzw. Modelldimension

Aus diesen Faktoren erwächst eine Vielzahl an Möglichkeiten, das Training des HMBM durchzuführen und damit die Notwendigkeit, verschiedene Trainingsläufe und Trainingsalgorithmen zu bewerten und miteinander zu vergleichen.

In der Literatur zu Clusterverfahren werden zu diesem Zweck verschiedene Clusterindizes vorgeschlagen (z. B. in [17]), die das Resultat einer Clusterung bewerten. Je nach Definition des Indexes wird dann die Clusterung als beste erachtet, die diese Maßzahl maximiert oder minimiert. In [23] werden verschiedene Indizes auch für die HMM-Clusterung untersucht. Dafür ist es unter anderem auch notwendig, einen Abstand zwischen verschiedenen Hidden-Markov-Modellen zu definieren. Bei wenig strukturierten Daten ergibt sich bei dieser Vorgehensweise jedoch häufig das Problem, dass die Clusterindizes keine klare Auskunft über die beste Clusterung geben können; allenfalls sind gewisse Tendenzen erkennbar.

Die beiden in dieser Arbeit vorgestellten Verfahren zur Clusterung von Bausparsequenzen liefern als Resultat deutlich mehr als eine reine Partitionierung der Trainingssequenzen. Die Hauptfunktion der trainierten Modelle besteht einerseits darin, die statistischen Eigenschaften der Trainingsmenge möglichst gut zu erfassen und andererseits darin, unbekannte und unvollständige Sequenzen zu klassifizieren und zu komplettieren. Daher liegt es nahe, diese beiden Aspekte in die Bewertung der Clusterung bzw. der trainierten Modelle einzubeziehen.

Die folgenden drei Größen werden im weiteren Verlauf dieses Kapitels zur Bewertung verschiedener Clusterungen herangezogen:

Zielfunktion: Die Betrachtung der bei der Clusterung verwendeten Zielfunktion bezüglich der Trainingsmenge ist sinnvoll, wenn es darum geht, bei unveränderter Trainingsmenge und unveränderter Modelltopologie verschiedene Clusterdurchläufe miteinander zu vergleichen. Die Zielfunktion misst, wie gut sich das Modell auf die Trainingsdaten spezialisiert hat. Durch eine Vergrößerung der Modelldimension ist es in der Regel möglich, diese Spezialisierung zu steigern. Daher ist die Zielfunktion bezüglich der Trainingsmenge kein gutes Maß zum Vergleich unterschiedlich dimensionierter Modelle. Sinnvoller ist es in diesem Fall, die Zielfunktion bezüglich einer dem trainierten Modell unbekannten Testmenge von Sequenzen zu berechnen. Ein Verfahren, in dem dieser Ansatz verfolgt wird, ist die im Abschnitt 5.4.2 beschriebene Monte-Carlo-Cross-Validierung.

BIC: Das „bayesian information criterion“ (BIC) ist eine Approximation des Bayes-Faktors und stellt eine Maßzahl dar, die die Likelihood von Modellparametern und die Dimension des Modells zur Beurteilung von Modellen gegeneinander abwägt. Dieses Kriterium wurde erstmals in [34] vorgeschlagen und wird im Abschnitt 5.4.1 auf das HMBM angewendet.

Prognosegüte: Unter Prognosegüte soll in diesem Zusammenhang die Fähigkeit verstanden werden, Anfangssequenzen zu vervollständigen. Dazu werden Sequenzen (der Trainingsmenge oder einer unbekannten Testmenge) an einer vorzugebenden Position abgeschnitten und die fehlenden Enden der Sequenzen werden vom trainierten Modell generiert. Ein Spezialfall dieses Vorgehens besteht darin, keine Anfangssequenzen vorzugeben und alle Symbole der Sequenzen vom Modell generieren zu lassen. Die Vorgehensweise zum Generieren von Sequenzen wird im Abschnitt 6.2 erläutert. Zur Beurteilung der Modellgüte werden aus den realen Sequenzen und den künstlich generierten Sequenzen bauspartechnisch relevante Zeitreihen extrahiert und miteinander verglichen. Hierbei ist zu beachten, dass nicht nur das Training der Modelle, sondern auch die Art und Weise, wie Sequenzen generiert werden, eine Rolle spielen. Die Prognosegüte wird im weiteren Verlauf dieses Kapitels und im Kapitel 6 mehrfach zur Bewertung von Modellen herangezogen.

Die Bewertung von Trainingsalgorithmen geschieht primär anhand der berechneten Modelle und somit durch die oben angeführten Kriterien. Ein zusätzliches Kriterium stellt die Laufzeit des eingesetzten Verfahrens dar. Liefern zwei Verfahren gleichwertige Modelle, so ist jenes Verfahren zu bevorzugen, welches in kürzerer Laufzeit zum gewünschten Ergebnis kommt. Vergleiche der benötigten Laufzeiten finden sich im Abschnitt 6.4.

5.2 Modelltraining und Clusterung

5.2.1 Einfachster Fall: ein Modell

Beim HMM-Training mit nur einem Modell geht es nicht darum, Sequenzen zu partionieren oder zu klassifizieren, sondern darum zu untersuchen, ob es möglich ist, die statistischen Eigenschaften der Sequenzen unter Verwendung nur eines HMMs abzubilden. In diesem Spezialfall sind der HMM-Cluster-Algorithmus und der MIX-HMM-Algorithmus identisch.

Modellrekonstruktion bei artifiziellen Daten
Zunächst soll analysiert werden, welche Zielfunktionswerte sich nach Durchführung des Baum-Welch-Trainings ergeben, wenn der Algorithmus mit unterschiedlichen Initialmodellen gestartet wird. Da bei realen Sequenzen das Optimum der Zielfunktion nicht mit Sicherheit ermittelt werden kann, ist es sinnvoll bei dieser Untersuchung mit künstlich generierten Sequenzen zu arbeiten. Wenn hinreichend viele künstliche Sequenzen generiert werden, ergibt sich die optimale Likelihood gerade mit dem Generatormodell. Somit lassen sich die mit dem Baum-Welch-Algorithmus gewonnenen Modelle an diesem bestmöglichen Wert messen. Außerdem ist es möglich, das Trainingsmodell mit dem Generatormodell zu vergleichen und festzustellen, ob der Trainings-Algorithmus in der Lage ist, das Modell aus den Sequenzen zu rekonstruieren.

Das verwendete Generatormodell stellt ein (willkürliches) Beispiel dar und steht in keinem Zusammenhang zu den Modellen zur Abbildung der Bausparsequenzen. Es enthält nur eine Übergangsklasse, besteht aus vier Zuständen und ist durch folgende Übergangsmatrix

$$A = \begin{pmatrix} 0.7 & 0.05 & 0.05 & 0.2 \\ 0.0 & 0.5 & 0.25 & 0.25 \\ 0.0 & 0.0 & 0.7 & 0.3 \\ 0.5 & 0.0 & 0.0 & 0.5 \end{pmatrix},$$

den Initialwahrscheinlichkeiten

$$\pi = \begin{pmatrix} 0.5 & 0.5 & 0.0 & 0.0 \end{pmatrix}$$

und den beiden Parametervektoren für die Ausgabe

$$\mu = \begin{pmatrix} 0.0 & 5.0 & 10.0 & 20.0 \end{pmatrix},$$

$$\sigma = \begin{pmatrix} 0.5 & 0.5 & 2.0 & 5.0 \end{pmatrix}$$

gegeben. Als Ausgabefunktion wird die Gauß-Verteilung eingesetzt. Mit diesem Modell wurden 10 000 ungewichtete, aus jeweils genau zehn Symbolen bestehende Sequenzen generiert. Diese Sequenzen wurden als Trainingsmenge beim Baum-Welch-Training mit zehn unterschiedlichen, zufällig generierten Initialmodellen eingesetzt. Im Gegensatz zum Generatormodell waren die Übergangsmatrizen der Zufallsmodelle jeweils vollständig verknüpft.

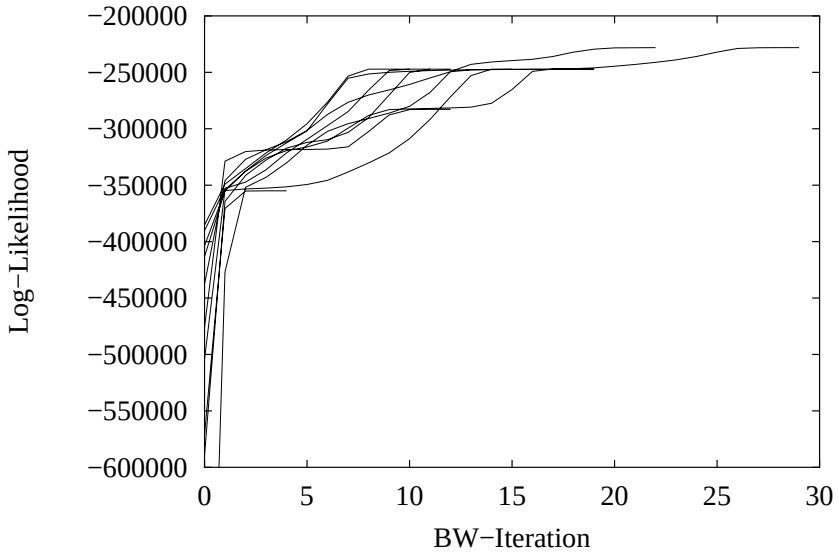

Abbildung 5.1: *Log-Likelihood im Baum-Welch-Algorithmus bei unterschiedlichen Initialmodellen*

In Abbildung 5.1 ist der Verlauf der Gesamtlikelihood

$$\begin{aligned} \log(L(\lambda)) &:= \log(p(O|\lambda)) \\ &= \sum_{k=1}^{K} \log(p(O^k|\lambda)) \end{aligned}$$

während der Baum-Welch-Iterationen für diese zehn Initialmodelle wiedergegeben. Es fällt auf, dass in diesem Fall der Algorithmus je nach Initialmodell gegen einen von vier verschiedenen Werten der Log-Likelihood-Funktion konvergiert. Der Wert von $\log(L(\lambda))$ im Fall des Generatormodells beträgt bei den 10 000 generierten Sequenzen –228 170. Dieser Wert wird in zwei der zehn Durchläufe exakt erreicht. Bei den übrigen acht Versuchen bleibt der Algorithmus jeweils in einem lokalen Maximum hängen und ist nicht in der Lage, den optimalen Zielfunktionswert zu erreichen. Somit besteht bei dieser einfachen Modelltopologie eine gute Chance, wenn auch keine Garantie, ein optimales Modell nach Durchführung mehrerer Trainingsläufe mit jeweils unterschiedlicher Initialisierung zu erhalten.

Nachfolgend sind die Parameter des vom Baum-Welch-Algorithmus ermittelten Modells aus ei-

nem dieser beiden Läufe mit der optimalen Log-Likelihood angegeben:

$$A = \begin{pmatrix} 0.4954 & 0.5046 & 0.0000 & 0.0000 \\ 0.1997 & 0.6973 & 0.0503 & 0.0526 \\ 0.2516 & 0.0000 & 0.5013 & 0.2471 \\ 0.2983 & 0.0000 & 0.0000 & 0.7016 \end{pmatrix},$$

$$\pi = \begin{pmatrix} 0.0000 & 0.4999 & 0.5001 & 0.0000 \end{pmatrix},$$

$$\mu = \begin{pmatrix} 19.9900 & -0.0003 & 5.0000 & 10.0020 \end{pmatrix},$$

$$\sigma = \begin{pmatrix} 4.9999 & 0.4989 & 0.4987 & 2.0172 \end{pmatrix}.$$

Das vom Training ermittelte Modell ist äquivalent zum Generatormodell. Lediglich die Zustände sind permutiert.

Anhand dieses konkreten Beispiels ist demonstriert worden, dass es mit dem Baum-Welch-Algorithmus in einfachen Fällen möglich ist,

- das globale Optimum der Zielfunktion zu erreichen und
- das Generatormodell aus den Trainingsdaten exakt zu rekonstruieren.

Dies wird dadurch erreicht, dass der Algorithmus mehrfach mit unterschiedlichen Initialisierungen gestartet wird.

Abbildung realer Bauspar-Sequenzen
Als nächstes soll nun die Abbildung realer Daten mit einem Hidden-Markov-Modell untersucht werden. In diesem Zusammenhang soll die Frage beantwortet werden, ob ein einzelnes Hidden-Markov-Modell in der Lage ist, eine gute Parametrisierung der statistischen Eigenschaften der Trainingsmenge zu liefern und welche Einfluss die Modelldimension hierbei hat.

Hierzu werden zwei getrennte Trainingsläufe mit unterschiedlicher Modelldimension und Modelltopologie in der Sparphase auf den realen Daten einer Tarifklasse eines kompletten Abschlussjahrganges (1985) durchgeführt:

1. Eine Topologie wie in Abbildung 4.3 auf Seite 68 dargestellt mit zwölf Zuständen wird vorgegeben.
2. Die Anzahl der Sparzustände wird gegenüber dem ersten Modell von vier auf zehn erhöht; damit enthält das Modell insgesamt 18 Zustände. Außerdem werden Zustandswechsel zwischen allen Sparzuständen zugelassen und die Markov-Kette kann in jedem der Sparzustände starten.

Mit den trainierten Modellen werden jeweils künstliche Sequenzen generiert und mit den Trainingssequenzen verglichen. Die Anzahl der Sequenzen ist in den drei Sequenzfeldern identisch und beträgt 12 429.

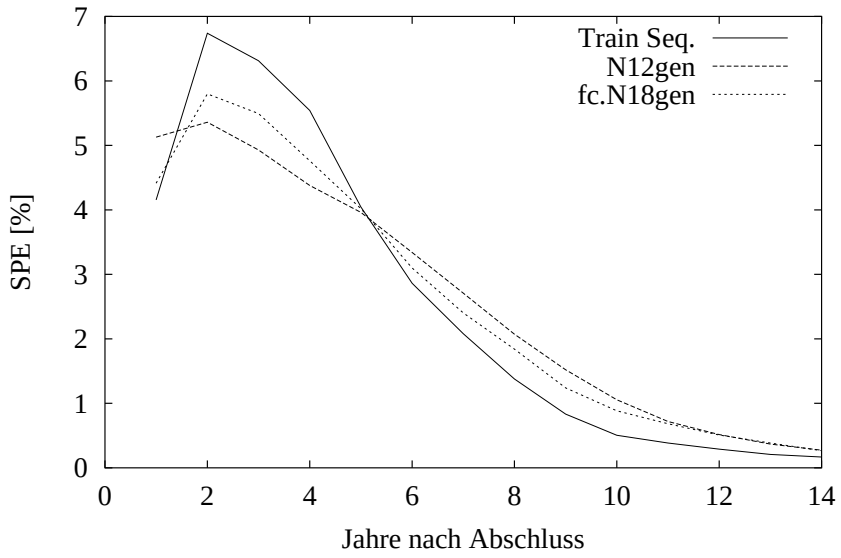

Abbildung 5.2: *Verlauf der jährlichen Spargeldeingänge eines Abschlussjahrganges bei realen und bei generierten Sequenzen (N = Anzahl der Zustände, fc = „fully connected“).*

In Abbildung 5.2 ist der mittlere jährliche Spargeldeingang in den einzelnen Jahren nach Vertragsabschluss der drei Sequenzfelder dargestellt. Die durchgezogene Linie gibt die Werte der realen Trainingssequenzen wieder. Aus der Abbildung ist ersichtlich, dass die allgemeine Tendenz des simulierten mit dem realen Spargeldeingang übereinstimmt.

Die Abstände zwischen den generierten und realen Spargeldeingängen sind im Modell mit 18 Zuständen deutlich geringer als im Modell mit nur zwölf Zuständen. Dies trifft auch auf die weiteren Kollektivzeitreihen, die hier nicht dargestellt sind, zu. Dieses Ergebnis kann dadurch erklärt werden, dass dem größeren Modell mehr Parameter zur Verfügung stehen, um sich an die Trainingsdaten anzupassen. Somit ist es auch besser in der Lage, eine Vielfalt von Verhaltensmustern abzubilden. Dies spiegelt sich auch in der mittleren Likelihood pro Trainingssequenz bei den beiden Modellen wider. Es ergibt sich:

$$\begin{aligned} \log(L(\lambda_{N12})) &= -13.36 \\ \log(L(\lambda_{fc.N18})) &= -10.31. \end{aligned}$$

Eine noch bessere Übereinstimmung der Zeitreihen könnte durch eine weitere Vergrößerung der Modelldimension erreicht werden.

Als Resümee dieser Untersuchungen kann festgehalten werden, dass eine gewisse Abbildungsqualität der Bauspar-Sequenzen mit nur einem Hidden-Markov-Modell erreicht werden kann, die sich durch eine Erhöhung der Modelldimension noch steigern lässt. Wie sich in den nachfolgenden Abschnitten zeigen wird, lässt sich unter Verwendung mehrerer Modelle und der Clusterverfahren MIX-HMM und HMM-Cluster die Übereinstimmung von Trainings- und gene-

rierten Sequenzen gegenüber dem Einsatz nur eines Modells noch deutlich verbessern. Außerdem können mehrere Modelle unterschiedliche Verhaltensweisen repräsentieren.

5.2.2 Initialisierungen

Die Clusterung der Sequenzen und das Training der Hidden-Markov-Modelle, sei es mit dem HMM-Cluster-Algorithmus (Abschnitt 2.3.5) oder mittels der Modellierung als Mischmodell (Abschnitt 3.2), führt im Allgemeinen auf Lösungen, die ein lokales Maximum der Zielfunktion darstellen. Keiner der beiden Algorithmen bietet die Garantie, das globale Optimum zu finden. Die resultierende Lösung hängt auf sehr komplexe Art und Weise von der gewählten Initialisierung ab. Daher ist es wichtig, eine möglichst geeignete Initialisierung zu finden. Als Maß für die Güte einer Initialisierung sollen hierbei der jeweilige Zielfunktionswert bezüglich der Trainingsmenge nach Durchführung der Clusterung herangezogen werden. Dies sind die Gleichungen (4.19) und (4.20), die zur besseren Übersicht nachfolgend erneut aufgeführt sind:

$$\begin{aligned} \text{HMM-Clusterung:} \qquad Z_{MD}(O,\lambda) &= \sum_{i=1}^{C} \sum_{k, O^k \in \mathcal{C}_i} w^k \log(p(O^k|\lambda_i)), \\ \text{MIX-HMM:} \qquad Z_{MIX}(O,\lambda,\alpha) &= \sum_{k=1}^{K} w^k \log\left(\sum_{i=1}^{C} \alpha_i p(O^k|\lambda_i)\right). \end{aligned} \tag{5.1}$$

Um verschiedene Initialisierungen miteinander zu vergleichen, ist es notwendig, alle weiteren Randbedingungen, die Einfluss auf die beiden obigen Gütekriterien haben, konstant zu lassen. Hierunter fallen die Zusammensetzung der Trainingsmenge und die Topologie der Modelle. Mit der Festlegung einer Topologie stehen die Anzahl der Modelle, die Anzahl der Zustände in den Modellen und die verbotenen Zustandswechsel fest.

Zur vollständigen Initialisierung müssen zum einen alle HMM-Parameter festgelegt werden. Zum anderen muss bestimmt werden, mit welcher Sequenzpartition der Algorithmus startet (HMM-Clusterung) bzw. welche a posteriori Komponentenwahrscheinlichkeiten (s. Gleichung (3.14)) den einzelnen Modellen bei Vorliegen der jeweiligen Trainingssequenz zugeordnet werden (MIX-HMM). Im Fall der Modellierung als Mischmodell muss außerdem noch die a priori Wahrscheinlichkeit α_i festgelegt werden.

Folgende Initialisierungen sollen untersucht werden:

Modellparameter

ZUF: Alle Parameter werden zufällig belegt, jedoch so, dass sich Startmodelle ergeben, die eine geringe Spezialisierung aufweisen. Hierzu muss die Varianz der Ausgabedichten hinreichend groß gewählt werden. Die dazugehörigen Mittelwerte sollten in der Größenordnung vieler Sequenzsymbole liegen.

GLEICH: Wie bei „ZUF“ werden alle Parameter zufällig belegt, jedoch für alle Modelle exakt gleich.

PROTO: In der Kenntnis bestimmter typischer, real vorkommender Verhaltensweisen werden Initialmodelle bereits zu Beginn so definiert, dass sie unterschiedliche dieser Verhaltensmuster abbilden.

Startpartition

SP_BEST: Jede Sequenz wird gemäß der Klassifikation dem jeweils besten Modell zugeordnet. Im Fall von MIX-HMM entspricht dies einer Festlegung der Komponentenwahrscheinlichkeit auf eins für das beste Modell und auf null für die übrigen Modelle.

SP_ZUF: Jede Sequenz wird zufällig einem Modell ohne Berücksichtigung des Klassifikators zugeordnet.

SP_VERT: Die a posteriori Komponentenwahrscheinlichkeiten werden basierend auf den Initialparametern mit dem rekursiven Algorithmus aus Abschnitt 3.2 berechnet und bereits für die Initialisierung herangezogen (nur MIX-HMM).

NO_SP: Ein abweichendes Verfahren besteht darin, im ersten Schritt auf eine Partitionierung der Sequenzen zu verzichten und jedes Modell mit der gesamten Trainingsmenge zu trainieren. Im Falle des MIX-HMM-Algorithmus entspricht dies einer gleichförmigen Belegung der Komponentenwahrscheinlichkeit.

Die aufgeführten Initialisierungen der Modellparameter und der Sequenzpartition können auf verschiedene Arten miteinander kombiniert werden, wobei nicht jede Kombination sinnvoll ist.

Die Tabellen 5.1 und 5.2 enthalten die Zielfunktionswerte der untersuchten Kombinationen. Bei Daten, die mit einem Stern versehen sind, handelt es sich um Mittelwerte, gemittelt jeweils über fünf verschiedene Durchläufe. Es ist zu beachten, dass die Werte für die beiden Verfahren nicht miteinander vergleichbar sind.

	ZUF	**GLEICH**	**PROTO**
SP_BEST	-1202412*	—	-885248
SP_ZUF	-976826*	-1525780*	—
NO_SP	-1064721*	—	-868502

Tabelle 5.1: *Zielfunktionswerte $Z_{MD}(O, \lambda)$ verschiedener Initialisierungen bei der HMM-Clusterung*

Der Grund dafür, dass die Werte von $Z_{MD}(O, \lambda)$ deutlich über denen von $Z_{MIX}(O, \lambda, \alpha)$ liegen, ist darin zu sehen, dass bei der Berechnung von $Z_{MD}(O, \lambda)$ die Wahrscheinlichkeitsdichten des jeweils besten Modells eingehen, während bei $Z_{MIX}(O, \lambda, \alpha)$ zumindest in Anteilen auch schlechtere Modelle zur Zielfunktion beitragen.

Bezüglich der Initialmodelle wird aus den ermittelten Zahlen deutlich, dass die Initialisierungen „ZUF“ und „PROTO“ recht gute und in etwa vergleichbare Log-Likelihood-Werte liefern, mit etwas besseren Ergebnissen bei der Initialisierung „PROTO“. Die Vorgabe von guten Prototypen

	ZUF	GLEICH	PROTO
SP_BEST	-1782803*	—	-1479335
SP_ZUF	-1496645*	-2045213*	—
SP_VERT	-1530497*	—	-1346723
NO_SP	-1394357*	—	-1386098

Tabelle 5.2: *Zielfunktionswerte $Z_{MIX}(O, \lambda, \alpha)$ verschiedener Initialisierungen beim MIX-HMM-Algorithmus*

setzt jedoch bereits fundierte Kenntnisse der zu modellierenden Daten voraus. Die Initialisierung mit Zufallsmodellen ist allgemeiner und wird daher bei den weiteren Untersuchungen eingesetzt.

Bei der Wahl der Startpartition ergeben sich etwa gleich gute Werte bei der zufälligen Startpartition und beim Verzicht auf eine Partitionierung in der ersten Iteration. Den nachfolgenden Ergebnissen dieser Arbeit liegt jeweils eine zufällige Startpartition zugrunde.

5.2.3 HMM-Cluster versus MIX-HMM

In den Abschnitten 2.3.5 und 3.2 wurden zwei unterschiedliche Verfahren – der HMM-Cluster-Algorithmus und der MIX-HMM-Algorithmus – zur Clusterung der Daten und zum Training der Hidden-Markov-Modelle vorgestellt. Hier soll es nun darum gehen, die beiden Methoden miteinander zu vergleichen, ihre Gemeinsamkeiten und Unterschiede darzulegen und ihre Eignung zur Simulation von Bausparkollektiven zu demonstrieren bzw. zu vergleichen.

Die beiden Verfahren unterscheiden sich in ihrem Modellierungsansatz, der sich in zwei unterschiedlichen Zielfunktionen niederschlägt, die es zu maximieren gilt (Gleichungen (4.19) und (4.20)).

Die Gemeinsamkeit der Algorithmen besteht darin, dass sie nach Vorgabe von Initialmodellen das Training der einzelnen Modelle basierend auf individuellen Sequenzfeldern unter Verwendung des Baum-Welch-Algorithmus durchführen.

Bei der HMM-Clusterung ändert sich mit jeder Iteration die Zusammensetzung der einzelnen Sequenzfelder durch den Austausch von Sequenzen. Die Sequenzen bilden eine Partition und das Training geschieht jeweils innerhalb einer solchen Teilmenge. Beim MIX-HMM-Algorithmus dagegen werden die Sequenzen nicht partitioniert. Jedes Modell wird mit der Gesamtmenge der Sequenzen trainiert, jedoch mit einer vom Modell abhängigen Gewichtung der einzelnen Sequenzen, die nach jeder Iteration neu errechnet wird. Aus Sicht der Sequenz bedeutet dies, dass bei der HMM-Clusterung jede Sequenz nur das im Sinne der Likelihood nahe liegendste Modell beeinflusst, während beim MIX-HMM-Verfahren ein Einfluss auf alle Modelle besteht, jedoch gewichtet mit der Wahrscheinlichkeit, dass die betrachtete Sequenz vom Modell erzeugt wurde. Dieser Umstand hat Auswirkungen auf die Laufzeit der beiden Methoden, die näher im Abschnitt 6.4 untersucht werden.

Der MIX-HMM-Algorithmus liefert neben den HMM-Parametern auch direkt die a priori Wahrscheinlichkeiten der einzelnen Modelle in Form der α_j aus Gleichung (3.12), die für das Generie-

ren künstlicher Sequenzen verwendet werden können. Beim HMM-Cluster-Algorithmus können analoge Größen in einem zweiten Schritt aus der Anzahl der den einzelnen Modellen zugeordneten Trainingssequenzen berechnet werden.

Es soll jetzt die Frage untersucht werden, mit welchem der beiden Modellansätze Bausparsequenzen angemessener beschrieben werden können. Zu diesem Zweck werden fünf zufällige, für beide Verfahren identische Initialmodelle vorgegeben. Mit diesen Startmodellen und allen Sequenzen einer Tarifklasse des Abschlussjahres 1985 werden beide Trainingsverfahren durchgeführt. Die daraus resultierenden trainierten Modelle werden zur Verlängerung unbekannter Testsequenzen eingesetzt. Hierzu werden alle Sequenzen der gleichen Tarifklasse wie die Trainingsdaten, aber mit dem Abschlussjahr 1986 verwendet, wobei alle Symbole nach 1990 entfernt werden. Mit diesem Vorgehen ist eine Situation gegeben, die mit den Gegebenheiten einer kompletten Kollektivsimulation vergleichbar ist, mit dem Unterschied, dass die generierten Daten direkt anhand der „wahren“ Daten bewertet werden können.

Für einen aussagekräftigen Vergleich der beiden Clusterverfahren ist es wichtig, dass die Entscheidung, mit welchem der zur Auswahl stehenden Modelle eine vorgegebene Anfangssequenz verlängert werden soll, in beiden Fällen nach dem gleichen Kriterium erfolgt. Auf die dabei möglichen Varianten wird im Abschnitt 6.1 eingegangen. Hier wird bei vorgegebener Sequenz O^k jenes Modell i verwendet, für das gilt:

$$i = \arg\max_j(\alpha_j p(O^k|\lambda_j)), \tag{5.2}$$

wobei im Fall der HMM-Clusterung α_j berechnet wird durch

$$\alpha_j = \frac{\text{Anz. Sequenzen in Modell } j}{K}. \tag{5.3}$$

Abbildung 5.3 zeigt beispielhaft die Zeitreihen der Zuteilungen, die sich aus dem oben beschriebenen Vorgehen ergeben. Bis 1990 verlaufen die drei Kurven identisch, da erst ab 1991 generierte Daten eingehen. Es wird deutlich, dass das MIX-HMM-Verfahren besser in der Lage ist, die Realzuteilungen zu reproduzieren. Versuche sowohl mit unterschiedlichen Initialisierungen (s. Abschnitt 5.2.2) als auch ohne Verwendung der oben berechneten α im Fall der HMM-Clusterung, ergeben ein ähnliches Bild.

Neben den Zuteilungen gibt es einige weitere Größen, die bei der Durchführung einer Kollektivsimulation relevant sind. In der Tabelle 5.3 sind für einige dieser Kollektivgrößen die über die prognostizierten Zeiträume summierten Betragsdifferenzen zwischen den realen und den generierten Daten aufgeführt. Hierin kennzeichnet „HMM-Cluster“ die Generierung der Sequenzen ohne Berücksichtigung der a priori Modellwahrscheinlichkeit α nach Gleichung (5.3) unter Verwendung des mit dem HMM-Cluster-Verfahren trainierten Modells und „HMM-Cl, prior“ gibt an, dass die gleichen Modelle, allerdings unter Berücksichtigung der α zugrunde liegen.

Aus den obigen Daten wird deutlich, dass die MIX-HMM-Clusterung die geeignetere Methode zur Clusterung der Bausparsequenzen darstellt. Jedoch ergibt sich dieser Vorteil auf Kosten einer höheren Laufzeit (s. Abschnitt 6.4).

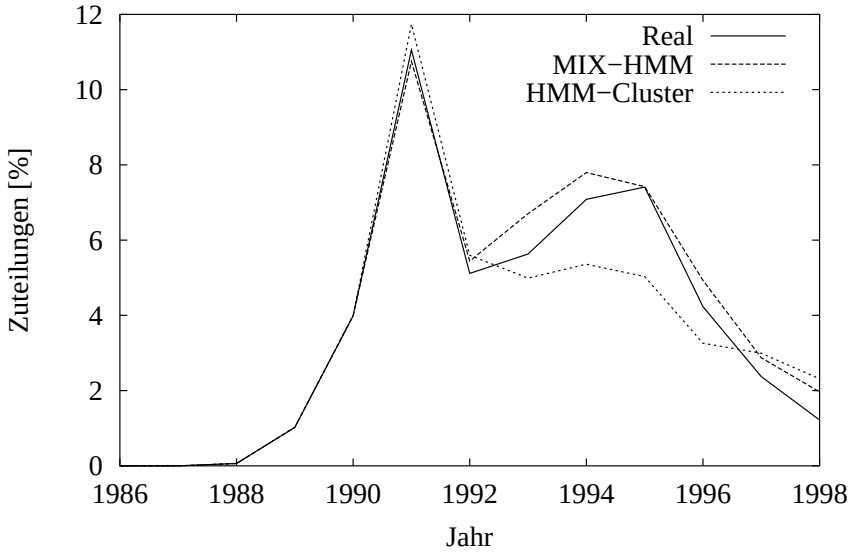

Abbildung 5.3: *Vergleich der Verfahren MIX-HMM und HMM-Cluster: reale und ab 1991 prognostizierte Zuteilungen des Abschlussjahrganges 1986.*

	MIX-HMM	HMM-Cluster	HMM-Cl, prior
SPE	0.866	1.033	0.993
Kündigung	7.691	6.736	7.187
Zuteilung	4.389	8.493	8.601
Darlehensverzicht	4.068	2.738	3.312
Fortsetzung	2.502	5.123	3.343
Tilgung	2.802	3.149	4.484
ungew. Summe	22.318	27.272	27.92

Tabelle 5.3: *Vergleich der hmm-basierten Clusteralgorithmen „MIX-HMM“ und „HMM-Cluster“*

5.2.4 Variation der Trainingsdaten

In diesem Unterkapitel sollen verschiedene Zusammensetzungen der Trainingsdaten untersucht werden. Zunächst wird demonstriert, wie das HMBM trainiert werden kann, wenn die zur Verfügung stehenden Trainingssequenzen in der Mehrzahl nicht den gesamten Ablauf vom Vertragsabschluss bis zur Vertragsabwicklung abdecken. Daran anschließend wird der Einfluss des Umfanges der Trainingsmenge untersucht.

Abschnittweises Training
In der Praxis der Bausparsimulationen stellt sich häufig das Problem, dass die vorhandenen Trainingsdaten nicht den gesamten Zeitraum eines Vertragsablaufes überdecken. Die Lösung dieser Problematik besteht darin, eine Trainingsmenge zu bilden, deren Sequenzen jeweils nur einzel-

ne Phasen des Ablaufs eines Bausparvertrages repräsentativ abdecken. Im Training des Modells wird dann sichergestellt, dass jede Trainingssequenz genau die Modellparameter beeinflusst, die mit der dazugehörigen Phase korrespondieren. Wie bereits in 4.3.2 erwähnt, besteht die Trainingsmenge aus zwei Gruppen:

Sparsequenzen: Sie enthalten ausschließlich Symbole für Spargeldeingang, Kündigung, Fortsetzung, Zuteilung und Darlehensverzicht.

Tilgungssequenzen: Sie beinhalten die Symbole Zuteilung, Tilgung und Tilgungsende.

Zur Bildung der Sparsequenzen werden all jene Verträge gewählt, die in 1985 abgeschlossen wurden. Die Tilgungssequenzen bestehen aus Verträgen, deren Tilgungsende im Jahr 1998 liegt, dem aktuellsten zur Verfügung stehenden Datenjahr. Für die Sparsequenzen ist das Zuteilungssymbol ein mögliches Zeichen, mit dem die Sequenz enden kann, während es für jede Tilgungssequenz den Anfang der Sequenz markiert. Aus dem zweiten Punkt ergibt sich, dass Modelle, die mit einem solchen Sequenzfeld trainiert werden, für den Zuteilungszustand eine nicht verschwindende Initialwahrscheinlichkeit erhalten. Zum Generieren neuer Sequenzen mit diesem Modell muss dieser Wert auf null gesetzt und die Initialwahrscheinlichkeiten der übrigen Zustände müssen entsprechend renormiert werden, da keine reinen Tilgungssequenzen ohne vorangestellte Sparphase erzeugt werden dürfen.

Zum Training des HMBM mit dieser zusammengesetzten Trainingsmenge sind zwei Anmerkungen zu machen:

1. Korrelationen, die zwischen Aktionen in der Sparphase und solchen in der Darlehensphase bestehen, gehen bei dieser Modellierung verloren.
2. Die Trainingsmenge stellt keinen realen Abschlussjahrgang dar, da die korrespondierenden Verträge aus unterschiedlichen Jahrgängen stammen. Dennoch sollen die trainierten Modelle u. a. dazu dienen, die Weiterentwicklung kompletter Abschlussjahrgänge zu prognostizieren. Auch dieser Punkt soll hier analysiert werden.

Zur Untersuchung des abschnittsweisen Trainings werden mit der oben angegebenen, zusammengesetzten Trainingsmenge fünf zufällig generierte Modelle mit einer Topologie gemäß Abbildung 4.3 mit dem MIX-HMM-Algorithmus trainiert. Zum Vergleich wird ein entsprechendes Training mit den gleichen Initialmodellen basierend auf einer Trainingsmenge, die aus allen Verträgen (der Tarifklasse 3) des Abschlussjahrganges 1985 besteht, durchgeführt.

Diese Modelle sollen hinsichtlich ihrer Möglichkeiten, Sequenzen zu prognostizieren, untersucht werden. Beispielhaft werden hierzu die Sequenzen des Abschlussjahrganges 1986 zum Jahr 1992 abgeschnitten. Die fehlenden Symbole werden mit den trainierten Modellen generiert, wobei die Wahl des Generatormodells erneut gemäß Gleichung (5.2) vorgenommen wird. Die so entstandenen Sequenzfelder werden mit den vorliegenden realen Daten verglichen. Hierbei ist zu bemerken, dass die Mehrzahl der zu komplettierenden Anfangssequenzen in keiner der beiden Trainingsmengen enthalten ist. Somit ist gewährleistet, dass keiner der Trainingsläufe gegenüber dem Vergleichslauf dadurch im Vorteil wäre, dass die Trainingsdaten und die zu verlängernden Daten teilweise oder ganz übereinstimmen.

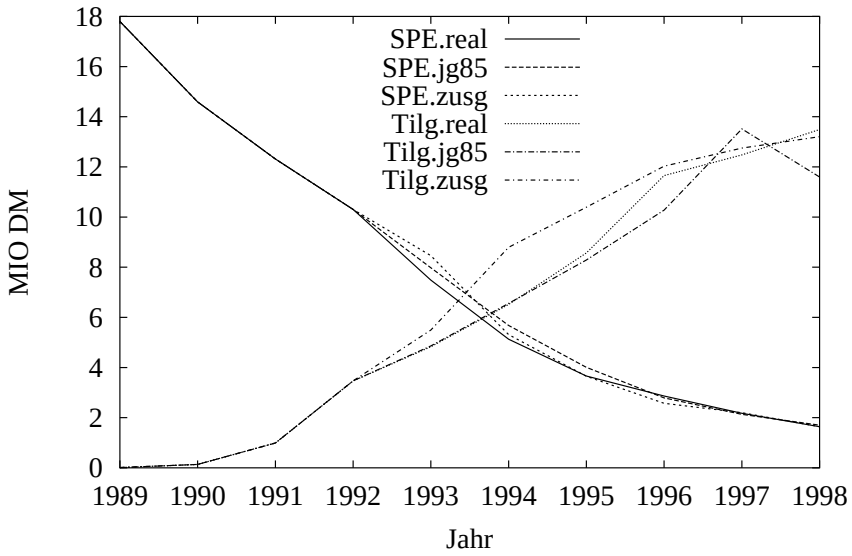

Abbildung 5.4: *Reale und ab 1993 prognostizierte Spargeldeingänge (SPE) und Tilgungszahlungen (Tilg) des Abschlussjahrganges 1986. Training mit zusammengesetzter Trainingsmenge und mit dem Abschlussjahrgang 1985.*

In der Abbildung 5.4 sind die jährlichen Sparzahlungen und die jährlichen Tilgungszahlungen der betrachteten Sequenzen wiedergegeben, wobei die übereinstimmenden Daten der Jahre 1986 bis 1988 aus Gründen einer größeren Übersichtlichkeit nicht dargestellt sind.

Im Fall der Sparzahlungen kann für beide Trainingsmengen eine gute Übereinstimmung von realen und künstlichen Daten festgestellt werden. Bezogen auf die Sparphase sind die Trainingsmengen identisch. Vorkommende geringe Abweichungen der Spargeldeingänge zwischen den beiden generierten Sequenzfeldern haben daher ihren Ursprung in der unterschiedlichen Zusammensetzung der Trainingsdaten bezogen auf die Darlehensphase. Basierend auf den minimalen Abweichungen der Sparzahlungen kann keiner der beiden Trainingsmengen der Vorzug gegeben werden. Die resultierenden Modelle sind gleichermaßen gut geeignet, die fehlenden Daten zu prognostizieren. Insbesondere die oben erwähnte Vernachlässigung von Korrelationen zwischen Spar- und Darlehensphase hat auf die Modellierung von Sparzahlungen wenig Auswirkungen.

Bei den Tilgungszahlungen ist festzustellen, dass die ausschließlich aus den Verträgen mit Abschluss in 1985 bestehende Trainingsmenge zu einer besseren Modellierung führt. Insbesondere in den ersten Jahren der Prognose (1993 - 1995) zeigt sich dort eine sehr gute Übereinstimmung. Die Abweichungen im Fall der zusammengesetzten Trainingsmenge weisen darauf hin, dass ein kompletter Abschlussjahrgang als Trainingsmenge vorzuziehen ist.

Als weitere wichtige Kollektivgrößen sind in Abbildung 5.5 die Bausparsummen der Darlehensnehmer des jeweiligen Jahres dargestellt. Als Darlehensnehmer eines Jahres werden hier alle Sequenzen gewertet, die in dem betreffenden Jahr in die Darlehensphase eintreten. Bei beiden

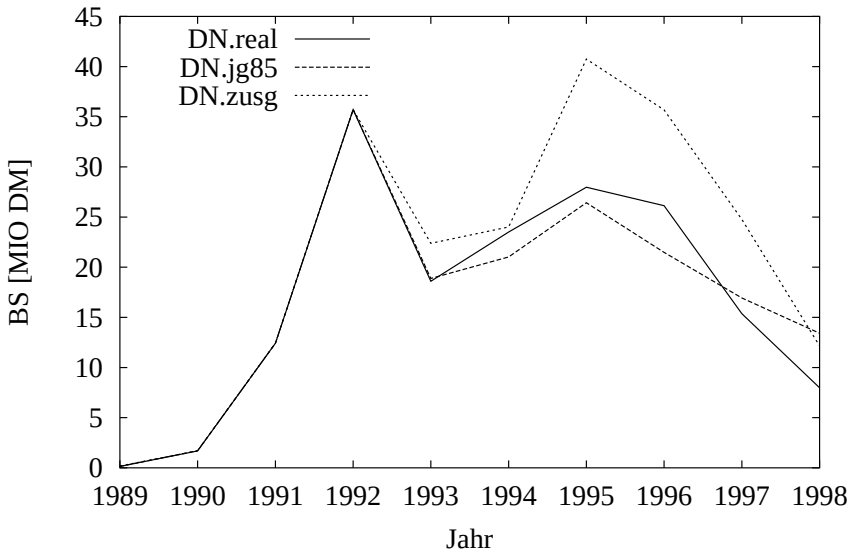

Abbildung 5.5: *Reale und ab 1993 prognostizierte Bausparsummen (BS) der Darlehensnehmer (DN) des Abschlussjahrganges 1986. Training mit zusammengesetzter Trainingsmenge und mit dem Abschlussjahrgang 1985.*

Trainingsmengen werden die realen Werte weniger gut getroffen als im Fall der Spargeld- und Tilgungszahlungen. Die zusammengesetzte Trainingsmenge liefert hier deutlich schlechtere Resultate. Eine Ursache hierfür könnte darin liegen, dass in der Trainingsmenge Sequenzen der Darlehenphase zu stark vertreten sind.

Zusammenfassend kann festgestellt werden, dass von den beiden untersuchten Trainingsmengen der vollständige Abschlussjahrgang vorzuziehen ist. Diese Aussage wird noch von weiteren, hier nicht dargestellten Kollektivzeitreihen untermauert. Dennoch kann es sinnvoll oder sogar notwendig sein, das Training auf eine zusammengesetzte Trainingsmenge zu basieren, wenn der Zeitraum, aus dem Daten vorhanden sind, noch kürzer ist als im untersuchten Fall.

Umfang der Trainingsmenge

Ein wichtiger Punkt beim Training des HMBM ist die Größe der Trainingsmenge. Generell ist es beim datenbasierten Training wichtig, dass der Umfang der Trainingsmenge nicht zu klein gewählt wird. Andernfalls repräsentiert das trainierte Modell die Trainingsmenge hinreichend gut, ist aber nur ungenügend in der Lage auf eine unbekannte Testmenge zu generalisieren. Somit stellt sich die Frage, wann eine Trainingsmenge hinreichend groß ist.

Bei sehr einfachen Problemen, wie z. B. der Schätzung der Parameter einer Normalverteilung, ist es möglich, Konfidenzintervalle in Abhängigkeit vom Umfang der Daten anzugeben. Damit kann die benötigte Datenmenge zum Erreichen einer bestimmten Sicherheit bei der Parameterschätzung bestimmt werden. Bei komplizierteren Modellen wie dem HMBM scheint dieses Unterfangen aussichtslos, da die Parameter der Ausgabefunktionen und die Parameter der Übergangsmatrizen auf komplexe Art und Weise zusammenhängen und außerdem die vorlie-

genden Zustände modellimmanent unbeobachtbar sind. Hinzu kommt die Problematik, dass der Trainingsalgorithmus im Allgemeinen nicht das globale Optimum findet.

Ein simples Verfahren, mit dem dennoch Aussagen zum Umfang der Trainingsdaten möglich sind, besteht darin, das HMBM mit unterschiedlich großen Datensätzen zu trainieren und mit den so gewonnenen Modellen die Likelihood bezüglich einer unbekannten Testmenge zu berechnen. In der Abbildung 5.6 wurde als Testmenge der Abschlussjahrgang 1997 gewählt und für alle Likelihood-Berechnungen festgehalten. Wieder wurden fünf Modelle der Topologie gemäß Abbildung 4.3 mit dem MIX-HMM-Verfahren trainiert unter Verwendung unterschiedlicher Datenmengen, deren Umfang auf der X-Achse der Abbildung aufgetragen ist. Die Trainingsmenge setzt sich dabei aus den Daten der Abschlussjahrgänge 1985 und 1986 zusammen. Die wiedergegebenen Likelihood-Werte sind Mittelungen über mehrere Trainingsläufe unterschiedlicher Initialisierung. Bei den großen Trainingssätzen (5000 und mehr Sequenzen) wurde jeweils über fünf, bei den kleineren Trainingsmengen jeweils über 15 Durchläufe gemittelt.

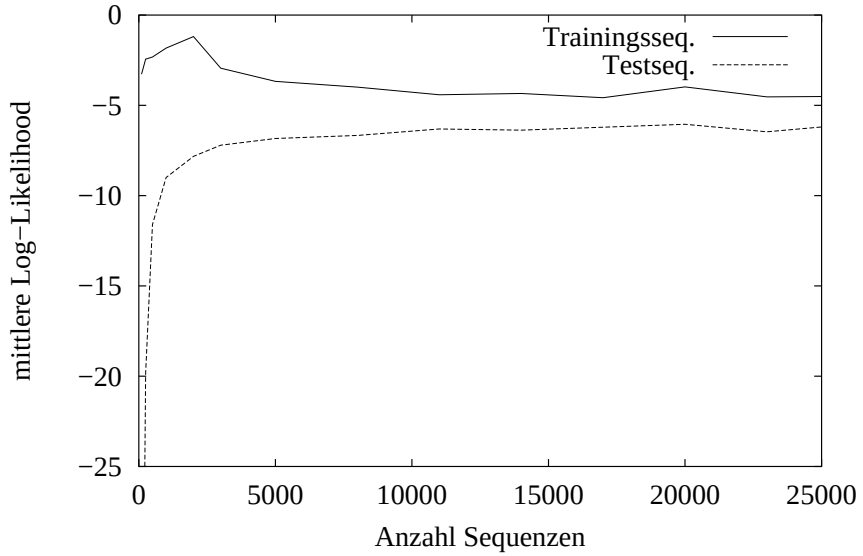

Abbildung 5.6: *Mittlere Log-Likelihood in Abhängigkeit vom Umfang der Trainingsmenge*

Die berechneten Log-Likelihood-Werte bezüglich der Testmenge lassen folgenden Schluss zu: Bei der verwendeten Modelldimension ist eine Trainingsmenge vom Umfang von etwa 3 000 – 5 000 Sequenzen notwendig und hinreichend. Eine darüber hinausgehende Vergrößerung der Trainingsmenge führt zu keiner Verbesserung der Abbildungsqualität. Trainingsmengen mit deutlich weniger als 3 000 Sequenzen liefern dagegen ein schwaches Ergebnis bezüglich der Testmenge.

Der Verlauf der mittlere Likelihood bezogen auf die Trainingsmenge ist bemerkenswert. Mit Abnahme der Sequenzanzahl steigt die mittlere Likelihood zunächst erwartungsgemäß an, da die Modelle sich auf die kleinere Trainingsmengen stark spezialisieren können. Unterhalb von etwa 2 000 kehrt sich jedoch der Trend um und die Likelihood fällt wieder. Bei sehr kleinen Trai-

ningsmengen sind je nach Initialisierung sehr starke Schwankungen der mittleren Likelihood zu beobachten, woraus gefolgert werden kann, dass in dem Fall die Existenz lokaler Optima ein gravierendes Problem darstellt. Eine Erklärung für die verhältnismäßig schlechte mittlere Likelihood könnte darin liegen, dass der Baum-Welch-Algorithmus bei sehr kleinen Trainingsmengen mit großer Wahrscheinlichkeit in einem relativ schlechten lokalen Optimum stecken bleibt.

5.2.5 Bausparsumme als Clustermerkmal

Im Abschnitt 3.3 wurden die Möglichkeiten erörtert, die Bausparsumme (BS) der Verträge als explizites Clustermerkmal in das HMM-Training einzubeziehen. Jede Sequenz setzt sich dabei aus den Symbolen der Zeitreihe und einer BS, die in Einheiten von 1000 DM (TDM) angegeben wird, zusammen, wobei die BS jeweils das erste Symbol einer Sequenz ist. Veränderungen der BS während der Vertragslaufzeit bleiben hier unberücksichtigt. Die Modellierung der BS geschieht durch die ebenfalls in Abschnitt 3.3 beschriebene Topologieerweiterung. Ein zusätzlicher Zustand mit einer Initialwahrscheinlichkeit von eins wird für die Ausgabe der BS eingeführt. Die Startwahrscheinlichkeit der übrigen Zustände ist folglich null. Vom Sonderzustand sind Übergänge in genau die Zustände erlaubt, für die in der Modellierung ohne BS eine Initialwahrscheinlichkeit größer null vorgesehen ist. Die Abbildung 5.7, in der der zusätzliche Zustand für die Bausparsumme mit BS und die Sparzustände mit S gekennzeichnet sind, veranschaulicht diese Modellerweiterung.

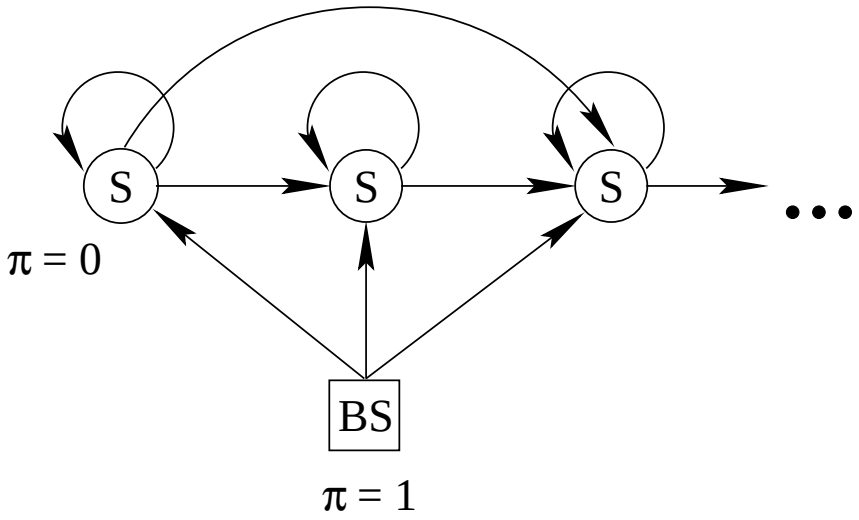

Abbildung 5.7: *Modellierung der Bausparsumme (BS) als Clustermerkmal durch eine Erweiterung der Modelltopologie um einen Startzustand zur Ausgabe der BS.*

Die aktuelle Implementierung der HMM-Bibliothek unterstützt keine Vermischung eines diskreten mit einem kontinuierlichen Hidden-Markov-Modell. Daher werden die diskreten Bausparsummen als kontinuierliche Werte interpretiert und ebenfalls durch die Normalverteilung im Modell parametrisiert. Verschiedene Versuche haben ergeben, dass es nicht ausreichend ist,

hierzu lediglich eine Verteilung pro Zustand einzusetzen. Bei nur einer Ausgabefunktion ergibt sich keine gute Überstimmung der realen Bausparsummenverteilung mit einer durch ein entsprechend trainiertes HMM generierten Verteilung. Eine Erhöhung der Anzahl der überlagerten Funktionen nach Gleichung (2.1) führt zu einer deutlichen Verbesserung. Abbildung 5.8 zeigt als Beispiel hierzu ein Histogramm der Bausparsummen der realen Sequenzen mit Abschlussjahr 1985 im Vergleich mit den Bausparsummen der generierten Sequenzen. Zur Ausgabe der generierten Bausparsumme wurden eine bzw. vier überlagerte Ausgabefunktionen pro Modell eingesetzt.

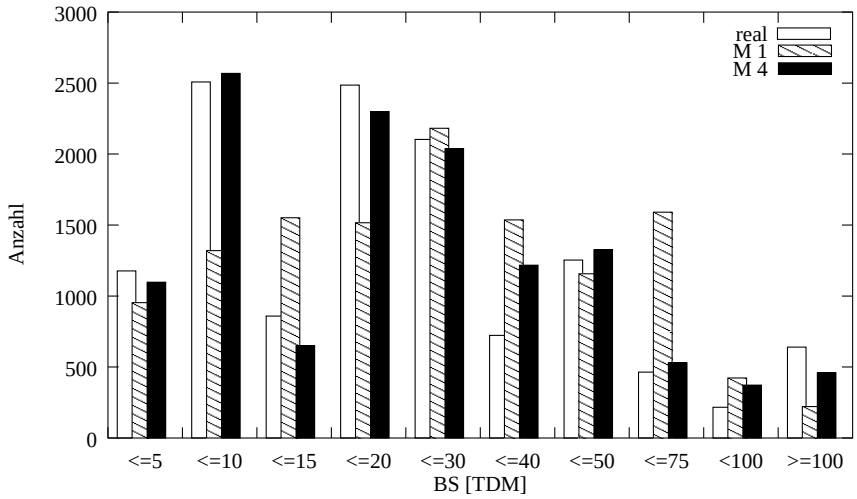

Abbildung 5.8: *Häufigkeiten von realen und generierten Bausparsummen mit einer Ausgabefunktion (M1) und mit vier Ausgabefunktionen (M4) des Abschlussjahrganges 1985*

Ziel der nachfolgenden Untersuchung ist es festzustellen, ob die BS ein relevantes Clustermerkmal ist. Liefert die Einbeziehung der BS Modelle, die in der Lage sind, die Eigenschaften der Sequenzen besser zu erfassen als Modelle, die ohne Berücksichtigung der BS gewonnen wurden? Diese Frage soll durch einen Vergleich der Trainingssequenzen mit den generierten Sequenzen untersucht werden. Hieraus ergibt sich folgendes Vorgehen:

1. Training geeigneter Initialmodelle unter Verwendung des vollständigen Abschlussjahrganges 1985 mit
   (a) Sequenzen, deren erstes Symbol die BS enthält. Diese Sequenzen erhalten keine zusätzliche Gewichtung mit der BS, andernfalls würde die BS quadratisch in die Clusterung eingehen, was eine absolut falsche Verteilung der BS in den generierten Sequenzen zur Folge hätte.
   (b) Sequenzen, die die Bausparsumme nicht als explizites Symbol, sondern lediglich als Gewichtung enthalten.
2. Generierung von Sequenzen in gleicher Anzahl wie die Trainingssequenzen mit
   (a) einer Gewichtung entsprechend der Ausgaben des BS-Zustandes.

(b) einer Einheitsgewichtung. Jede generierte Sequenz ist gleich bedeutsam.

3. Vergleich relevanter Kollektivgrößen der drei Sequenzfelder.

Die Tabelle 5.4, in der mit „Gen. keine BS“ generierte Einheitssequenzen ohne BS-Gewichtung und mit „Gen. BS Merkm.“ generierte Sequenzen mit BS-Gewichtung bezeichnet werden, gibt einige Ergebnisse aus obigem Vorgehen unter Verwendung von drei Ausgabefunktionen für die Bausparsumme wieder:

	Real	Gen. keine BS	Gen. BS Merkm.
Kündiger Ant.	30.93 %	**33.26 %**	39.55 %
Darl. Verz. Ant.	4.16 %	**6.12 %**	8.61 %
Forts. Ant.	10.59 %	13.31 %	**12.74 %**
Zuget. Ant.	51.21 %	63.34 %	**57.16 %**
mittl. Zut.	6.64 J.	9.35 J	**8.23 J**

Tabelle 5.4: *Bausparsumme als explizites Clustermerkmal: Einfluss auf wichtige Kollektivgrößen (Ant. = Anteil bezogen auf die gesamte BS in Prozent, Darl. Verz. = Darlehensverzichter, Forts. = Fortsetzer, Zuget. = zugeteilte Verträge, mittl. Zut. = mittlere Dauer bis zur Zuteilung in Jahren)*

Die Werte, die eine bessere Übereinstimmung mit den Realwerten darstellen, sind in der Tabelle fett gekennzeichnet. Aus diesen Zahlen kann geschlossen werden, dass für bestimmte Kollektivgrößen eine verbesserte Übereinstimmung mit den Realwerten erzielt werden kann, wenn die BS explizit in die Clusterung einbezogen wird, während für andere Größen genau des Gegenteil festzustellen ist. Auffällig ist eine starke Abweichung in der Dauer bis zur Zuteilung, unabhängig von der gewählten Clustermethode. Dies kann dadurch erklärt werden, dass in den Realdaten sehr späte Zuteilungen nicht vorkommen, da die Sequenzen maximal 14 Zeiträume umfassen, während in den generierten Sequenzen auch deutlich spätere Zuteilungen enthalten sind.

In der Abbildung 5.9 sind die Zuteilungen nicht in Abhängigkeit des Zeitpunktes, sondern in Abhängigkeit der erreichten Summe der Spargeldeingänge zum Zeitpunkt der Zuteilung für die drei Sequenzfelder dargestellt. Es ist festzustellen, dass die Clusterung mit BS nicht in Lage ist, den ausgeprägten Peak bei 40 % zu reproduzieren. Ein ähnliches Problem ergibt sich bei der (hier nicht dargestellten) zeitlichen Verteilung der Zuteilungen: bei den Realsequenzen ist dort ein deutlicher Peak nach sechs Jahren zu beobachten, der mit der BS-Clusterung nicht, mit der herkömmlichen Clusterung aber recht gut reproduzierbar ist.

Eine genauere Analyse hat gezeigt, dass die Ursache dieser Abweichungen darin zu sehen ist, dass im Fall der BS-Clusterung nicht genügend so genannte Soforteinzahler-Sequenzen generiert werden. Hierbei handelt es sich um Sequenzen mit einem SPE von 40 % im ersten Jahr und 0% in den Folgejahren, die in der Regel im sechsten Jahr das Zuteilungssymbol enthalten. Dieses Muster ist bei jenen Realsequenzen besonders häufig anzutreffen, die Verträge mit besonders hoher BS (bis zu mehrere Millionen DM) repräsentieren.

Wird die BS als Sequenzgewichtung eingesetzt, so führen diese gewichtigen Verträge zu Modellen, die das oben erwähnte Muster mit einer Häufigkeit generieren, die in Übereinstimmung mit

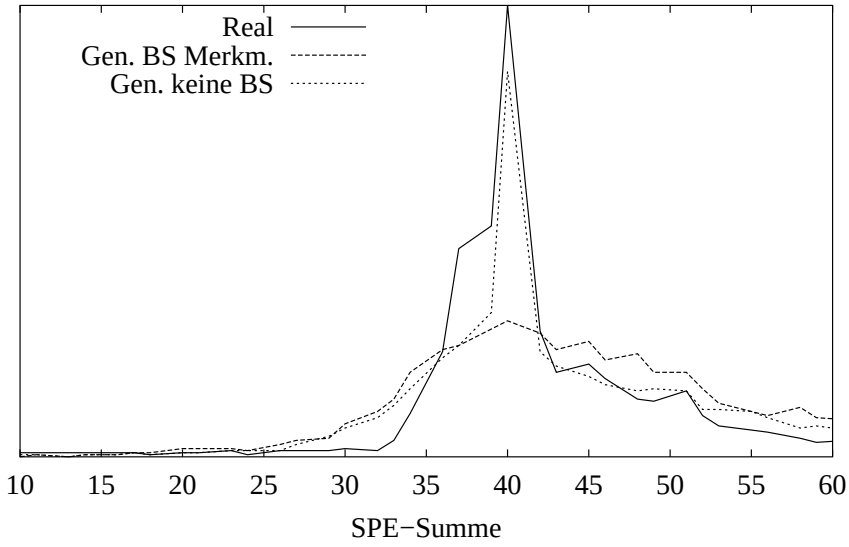

Abbildung 5.9: *Summe der Spargeldeingänge in Prozent der BS zum Zeitpunkt der Zuteilung beim realen Abschlussjahrgang 1985 und bei den generierten Sequenzen mit BS-Merkmal (Gen. BS Merkm.) und ohne BS-Merkmal (Gen. keine BS)*

der realen BS dieses Musters ist. Wird dagegen die BS als explizites Merkmal in ungewichteten (s. o.) Sequenzen eingesetzt, kann die positive Korrelation zwischen einer hohen BS und dem Sparverhalten „Soforteinzahler“ verloren gehen. Gleiches gilt auch für andere Verhaltensmuster.

Somit bleibt festzustellen, dass bislang wenig Argumente gefunden werden konnten, die für die Verwendung der BS als explizites Clustermerkmal sprechen. Eine Verbesserung könnte sich jedoch durch eine veränderte Definition der Übergangsklassen für die explizite BS-Modellierung ergeben, zu deren Festlegung allerdings noch weitergehende Untersuchungen nötig wären.

5.3 Lokale Extrema

Bei der Vorstellung des Baum-Welch-Algorithmus (Abschnitt 2.3.3) wurde erläutert, dass das Training der Modellparameter im Allgemeinen nicht das globale Optimum liefert, dass also die gefundene Lösung von den Anfangsbedingungen abhängt. Dabei kann sich bei unglücklicher Initialisierung durchaus eine sehr schlechte Lösung ergeben, die nicht akzeptiert werden sollte. In diesem Abschnitt werden drei verschiedene Ansätze zur Lösung dieser Problematik untersucht. Im ersten Teil werden die Verfahren beschrieben, die im zweiten Teil anhand einer geeigneten Testinstanz untersucht werden.

5.3.1 Methoden zur Überwindung lokaler Extrema

Die drei folgenden Verfahren sollen in dieser Arbeit zur Lösung der Problematik lokaler Optima untersucht werden:

verschiedene Initialmodelle: Die nahe liegendste Art, mit dieser Problematik umzugehen, besteht darin, mehrere Baum-Welch Trainingsläufe mit unterschiedlicher Initialisierung zu starten und das trainierte Modell mit dem größten Zielfunktionswert zu speichern.

erweiterter Baum-Welch-Algorithmus: Der Baum-Welch-Algorithmus wird dahingehend abgewandelt, dass ein vorzeitiger Abbruch in einem lokalen Maximum vermieden wird.

Simulated Annealing: Die aus der kombinatorischen Optimierung bekannte Methode des Simulated Annealing wird auf die Parameterschätzung in Hidden-Markov-Modellen angepasst.

Die erste Methode bedarf keiner weiteren Erläuterung, während der erweiterte Baum-Welch-Algorithmus und das Simulated Annealing nun näher beschrieben werden.

Erweiterter Baum-Welch-Algorithmus
Der Kern der Erweiterung des Baum-Welch-Algorithmus besteht darin, bei Eintritt der Konvergenz und damit bei Erreichen des (lokalen) Optimums, die aktuellen Modellparameter zu stören, um somit zu einem benachbarten Punkt im Parameterraum zu gelangen. Dieser benachbarte Punkt wird dann als Ausgangspunkt weiterer Baum-Welch-Iterationen genutzt, mit dem Ziel, durch diese Initialisierung ein gegenüber dem aktuellen Optimum verbessertes, neues Optimum zu erhalten. Wird die Zielfunktion als Energiefunktion interpretiert, so besteht die Idee, die hinter dieser Vorgehensweise steht darin, dass es durch die Fortsetzung der Baum-Welch-Iterationen in einem Nachbarpunkt des aktuellen Optimums möglich ist, lokale Barrieren der Energielandschaft zu überwinden.

Zur Darstellung des Algorithmus werden die beiden nachfolgenden Notationen verwendet:

1. **BW**(λ, O) bezeichnet den HMM-Parameter der sich aus der Durchführung des Baum-Welch-Algorithmus auf der Sequenzmenge O unter Verwendung des Initialmodells λ ergibt.
2. **Nachbar**(λ, ε) wählt im Parameterraum zufällig einen Punkt $\tilde{\lambda}$ aus der Nachbarschaft von λ. Die Größe des Nachbarschaftsraumes wird dabei vom Parameter $\varepsilon = (\varepsilon_a, \varepsilon_\pi, \varepsilon_\mu, \varepsilon_\sigma)$ festgelegt. Wie die Festlegung von $\tilde{\lambda}$ genau geschieht, wird nach Angabe des Algorithmus erläutert.

Mit diesen beiden Notationen lässt sich die Erweiterung zur Überwindung lokaler Maxima wie folgt angeben:

Erweiterter Baum-Welch-Algorithmus mit variabler Nachbarschaft

---

Eingabe: Startmodell λ_0, Sequenzfeld O, maximale Iterationszahl I,
Startumgebungsgröße ε_0, Levelumfang L, Wachstumsfaktor $W > 1$

$\hat{\lambda} := \mathbf{BW}(\lambda_0, O), \quad f := 0, \quad \varepsilon := \varepsilon_0$
for $i := 0$ **to** I **do**
1. $\tilde{\lambda} := \mathbf{Nachbar}(\hat{\lambda}, \varepsilon)$
2. **if** $(p(O|\hat{\lambda}) < p(O|\mathbf{BW}(\tilde{\lambda}, O)))$ **then**
   $\hat{\lambda} := \mathbf{BW}(\tilde{\lambda}, O)$
   $f := 0, \quad \varepsilon := \varepsilon_0$
   else
   $f := f + 1$
   end if
3. **if** $(f > L)$ **then**
   $\varepsilon := W \cdot \varepsilon$
   $f := 0$
   end if

end for

Ausgabe: Trainiertes Modell $\hat{\lambda}$

Abbildung 5.10 veranschaulicht die Idee des Algorithmus für den einfachen Fall, dass λ nur eindimensional ist. Die Indizes bei $\hat{\lambda}$ beziehen sich auf den Iterationszähler. Es sei angemerkt, dass der Vergleich zweier Modelle lediglich zwischen trainierten Modellen angestellt wird. Die Likelihood der zufällig gewählten Modelle spielt also keine unmittelbare Rolle im Gegensatz zum Simulated Annealing, das im Anschluss erläutert wird.

Die im Algorithmus eingehende Nachbarschaftsumgebung besitzt eine variable Größe. Nach Erreichen eines neuen Optimums wird zunächst in einer kleinen Nachbarschaft des Optimums nach einer Verbesserung gesucht. Führt dies nach L Versuchen zu keinem Erfolg, wird der die Größe der Nachbarschaft festlegende Parameter ε um den Faktor $W > 1$ erhöht. Hierdurch wird der Suchraum für einen neuen Initialparameter vergrößert. Wird die Nachbarschaft sehr groß gewählt, so ist das Verfahren äquivalent zur ersten Methode, der Durchführung multipler Trainingsläufe mit zufälliger Initialisierung. Bei Auffinden eines besseren Optimums wird die Größe des Suchraumes jeweils wieder auf den minimalen Wert ε_0 gesetzt.

Da ein neuer Parameter $\hat{\lambda}$ nur akzeptiert wird, wenn er eine Verbesserung der Zielfunktion bedeutet, kann diese Erweiterung niemals zu einer schlechteren Lösung als der konventionelle Baum-Welch-Algorithmus führen.

Die Implementierung des obigen Algorithmus erfordert die Berechnung neuer Modellparameter $\tilde{\lambda}$ in der Nachbarschaft eines vorgegebenen Parameters λ. Dieser neue Parameter $\tilde{\lambda}$ muss die beiden folgenden Randbedingungen erfüllen:

1. Die in $\tilde{\lambda}$ enthaltenen Übergangsmatrizen $\tilde{A}$ und die initiale Zustandswahrscheinlichkeit $\tilde{\pi}$ müssen den üblichen stochastischen Nebenbedingungen genügen. Für alle Parameter gilt, dass sie die Grenzen des Parameterraumes nicht verlassen dürfen.
2. Die Topologie des Modells darf nicht verändert werden. Das bedeutet insbesondere, dass Nullübergänge der Transitionsmatrix nicht verändert werden dürfen.

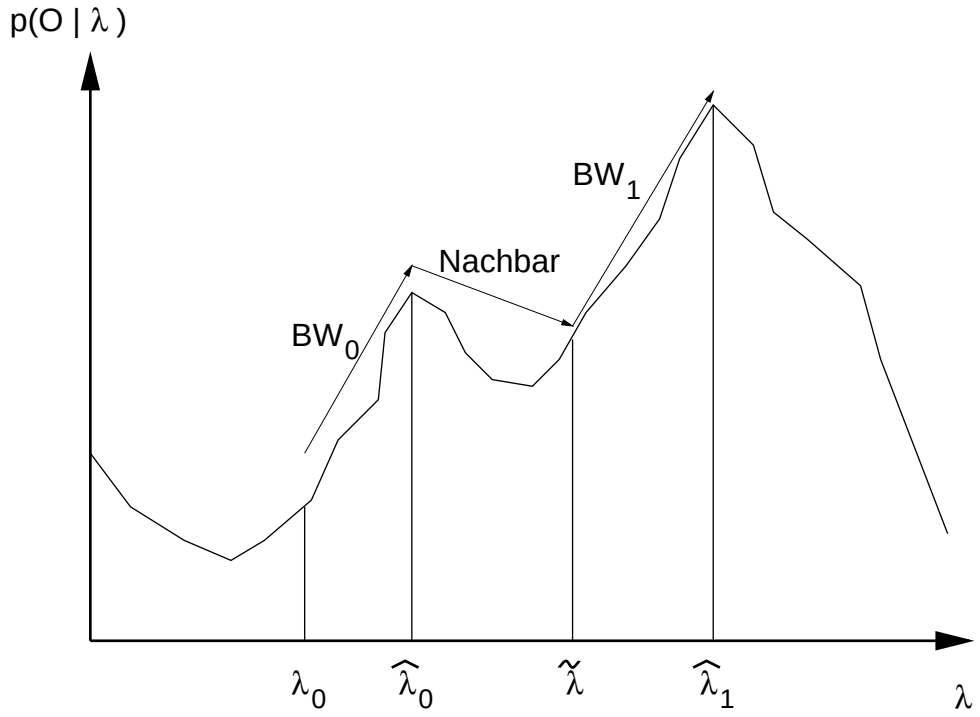

Abbildung 5.10: *Der erweitertete Baum-Welch-Algorithmus zur Überwindung lokaler Maxima*

Eine einfache, die beiden obigen Randbedingungen einhaltende Möglichkeit, einen Nachbarparameter zu finden, ist für einen Eintrag der Übergangsmatrix a_{ij} (vorausgesetzt $a_{ij} > 0$) durch folgende Berechnung gegeben:

$$\tilde{a_{ij}} := \frac{a_{ij} + \varepsilon_a \text{rand}_j(0,1)}{1 + \varepsilon_a \sum_{l:a_{i,l}>0} \text{rand}_l(0,1)},$$

wobei rand$(0,1)$ eine Zufallszahl im Intervall $[0,1]$ ist und ε_a der vorgegebene Parameter, der die Größe des Nachbarschaftsraumes bestimmt. Die Summation im Nenner geht ausschließlich über jene l, für die gilt: $a_{il} > 0$. Für π_i kann die obige Formel analog verwendet werden. Für die Parameter der Ausgabefunktionen (μ, σ) müssen keine Normierungsbedingungen eingehalten werden, lediglich der erlaubte Wertebereich darf nicht verlassen werden. Außerdem muss beachtet werden, dass der korrespondierende Zustand kein Sonderzustand ist, dessen Parameter vom Trainingsalgorithmus nicht verändert werden dürfen. Für $\tilde{\mu}$ wurde folgende Berechnung implementiert:

$$\tilde{\mu_i} := \mu_i + \text{rand}(-\varepsilon_\mu, \varepsilon_\mu).$$

Da der Wertebereich für μ im Fall von Sparzahlungen auf den Bereich $[0,100]$ und im Fall von Tilgungen auf das Intervall $[0,60]$ eingeschränkt ist (siehe Abschnitt 4.3), wird bei Über- oder Unterschreiten dieser Grenzen $\tilde{\mu_i}$ auf den Rand des erlaubten Intervalls gesetzt. Die Berechnung von $\tilde{\sigma}$ geschieht analog, wobei die gewonnenen Werte strikt positiv sein müssen.

Simulated-Annealing
Beim Simulated-Annealing handelt es sich um ein sehr allgemeines Optimierungsschema, das zum ersten Mal in [22] vorgestellt wird. Es basiert auf einer Analogie zwischen der statistischen Mechanik (also der Physik eines Vielteilchensystems im thermischen Gleichgewicht bei endlicher Temperatur) und der kombinatorischen Optimierung. Simulated Annealing ist nicht auf kombinatorische Optimierungsprobleme beschränkt, sondern kann auch auf die Optimierung von Funktionen kontinuierlicher Variablen ausgedehnt werden [16]. Das Verfahren wird nachfolgend sehr knapp und im HMM-Kontext dargestellt. Eine allgemeinere und ausführlichere Behandlung diese Themas ist z. B. in [1] zu finden.

Simulated Annealing auf Hidden-Markov-Modelle angewandt besteht im Wesentlichen aus den folgenden Elementen:

- Ausgehend von einem Initialmodell wird durch iterative Anwendung der Funktion **Nachbar**(λ, ε) aus der Baum-Welch-Erweiterung eine kontinuierliche Markov-Kette bestehend aus Hidden-Markov-Modellen gleicher Topologie definiert.
- Berechnung einer Akzeptanz-Funktion bei festgelegtem Paramter $T > 0$ (Temperatur):

$$SIMA(\tilde{\lambda}, \lambda, O) := \exp\left(\frac{\log(L_O(\tilde{\lambda})) - \log(L_O(\lambda))}{T}\right). \tag{5.4}$$

- Wird durch Anwendung von **Nachbar**(λ, ε) ein Nachbarmodell $\tilde{\lambda}$ von λ gefunden, für das gilt

$$\log(L_O(\tilde{\lambda})) > \log(L_O(\lambda)),$$

  so wird $\tilde{\lambda}$ als neues Ausgangsmodell akzeptiert: $\lambda := \tilde{\lambda}$. Andernfalls wird es mit der Wahrscheinlichkeit gegeben durch $SIMA(\tilde{\lambda}, \lambda, O)$ akzeptiert und entsprechend mit der Wahrscheinlichkeit $1 - SIMA(\tilde{\lambda}, \lambda, O)$ abgelehnt. Im Fall der Ablehnung wird λ beibehalten.
- Der Parameter T in (5.4) wird im Lauf der Iterationen von einem hohen Initialwert monoton verringert. Die Art und Weise, wie dies geschieht, definiert das so genannte Kühlschema[1] des Algorithmus und wird im nächsten Abschnitt erläutert.

Eine wesentliche Eigenschaft des Simulated Annealing liegt darin, dass eine im Vergleich zur aktuellen Lösung schlechtere Lösung mit Wahrscheinlichkeit *SIMA* als Ausgangspunkt für weitere Iterationen akzeptiert wird. Dies verhindert, dass der Algorithmus in einem lokalen Optimum verharrt, wenn in der Nachbarschaft kein besseres Modell gefunden werden kann. Die drei vorgeschlagenen Ansätze, mit der Problematik lokaler Optima umzugehen, werden im folgenden Abschnitt anhand einer Testinstanz miteinander verglichen.

[1] Die Bezeichnung Kühlschema und der Name „Simulated Annealing“ haben ihren Ursprung in der Analogie zum Abkühlen einer Schmelze. Wird die anfänglich heiße und ungeordnete Schmelze hinreichend langsam abkühlt, kann sie eine geordnetete Konfiguration (Kristall) auf niedrigstem Energieniveau annehmen (entsprechend dem globalen Optimum). Kühlt die Schmelze hingegegen zu schnell, so wird ein so genannter metastabiler Zustand eingenommen, der energetisch deutlich höher liegt (entsprechend dem lokalen Optimum).

5.3.2 Vergleich und Bewertung der Methoden

Ein wichtiger Aspekt beim Vergleich der drei Methoden zur Überwindung lokaler Maxima, ist die Frage nach der Laufzeit, die zum Erreichen einer bestimmten Likelihood notwendig ist. Bei hinreichend langer Rechendauer sind alle drei Methoden in der Lage, eine sehr gute Lösung zu finden. In der praktischen Anwendung hat diese Aussage jedoch keine Relevanz. Hier ist es interessant, wie lange eine Verfahren benötigt, um ein gefordertes Resultat zu erzielen. Daher werden bei der nachfolgenden Untersuchung die Likelihood-Werte als Funktion der Rechenzeit betrachtet.

Folgende Bedingungen liegen dem Vergleich zugrunde: ein einzelnes Modell mit 18 Zuständen, von denen die Sparzustände vollständig untereinander verbunden sind (s. auch Abschnitt 5.2.1), wird mit 5000 Bausparsequenzen des Abschlussjahres 1985 trainiert. Die Maximale Iterationszahl I wird beim erweiterten Baum-Welch-Algorithmus auf 80 gesetzt und die Anzahl der unterschiedlichen Initialmodelle beim wiederholten Baum-Welch Training wird auf 40 festgelegt. Hieraus ergeben sich in etwa vergleichbare Laufzeiten, da für die Durchführung eines Trainingslaufes im erweiterten Baum-Welch-Algorithmus im Mittel deutlich weniger Iterationen benötigt werden. Der Ursache hierfür ist darin zu sehen, dass die Ausgangsmodelle bereits teilweise trainiert sind. In der Abbildung 5.11 ist für jeweils fünf solcher Trainingsläufe die beste erzielte Log-Likelihood als Funktion der Realzeit in Sekunden wiedergegeben.

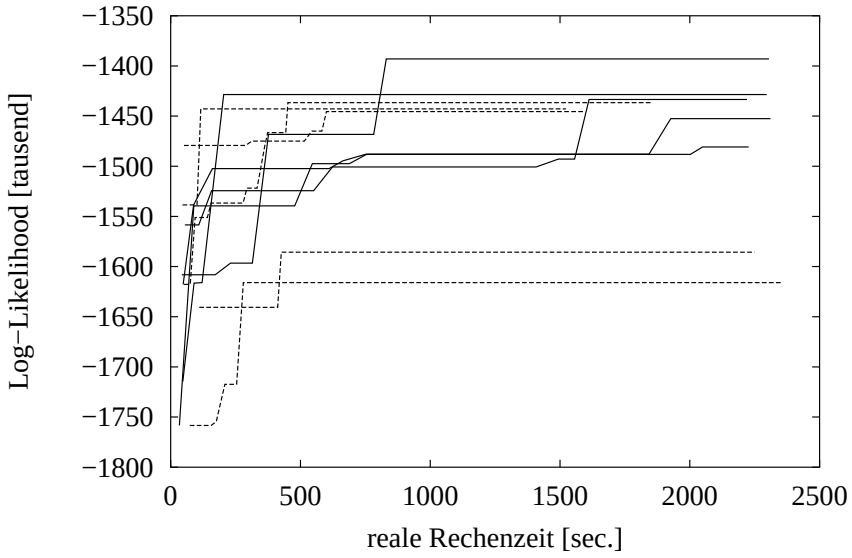

Abbildung 5.11: *Erzielte Log-Likelihood mit verschiedenen Initialmodellen (durchgezogene Linien) und mit dem erweiterten Baum-Welch-Algorithmus (gestrichelte Linien)*

In den betrachteten Trainingsläufen liefern beide Verfahren in etwa vergleichbare Zielfunktionswerte. Trotz der wachsenden Umgebungsgröße, innerhalb der nach einem verbesserten Modell gesucht wird, bleibt der erweiterte Baum-Welch-Algorithmus in den meisten Fällen sehr

früh in einem suboptimalen Modell stecken. Dies ließe sich durch eine deutliche Vergrößerung des Suchraumes vermeiden, dann jedoch geht das Verfahren in das wiederholte Baum-Welch-Training mit verschiedenen Initialmodellen über. Eine weitere, nicht untersuchte Möglichkeit, dies zu verhindern, bestünde darin, in Anlehnung an das Simulated Annealing mit einer gewissen Wahrscheinlichkeit auch schlechtere Modelle zu akzeptieren. Insgesamt kann festgestellt werden, dass der erweiterte Baum-Welch-Algorithmus gegenüben der simplen unterschiedlichen Initialisierung des Trainings keine Vorteile bietet.

Das Simulated Annealing wird mit gleicher Trainingsmenge und Modelltopologie wie das Baum-Welch Training durchgeführt. Untersuchungen mit verschiedenen Initialmodellen sind hier nicht wiedergegeben, da sich herausgestellt hat, dass die Initialisierung beim Simulated Annealing nur eine untergeordnete Rolle spielt. Sehr großen Einfluss hat dagegen die Wahl des Kühlschemas. Bei den nachfolgend dargestellten Untersuchungsergebnissen wurde folgendes Kühlschema verwendet:

$$T(t) = T_0 \cdot \exp\left(-\frac{\left(100 + \text{floor}(t/100) \cdot \Delta\right)^2}{66\,000}\right). \tag{5.5}$$

Hierbei bezeichnet t den Iterationszähler, T_0 und Δ sind frei wählbare Parameter. Die Funktion $\text{floor}(t/100)$ bewirkt, dass die Temperatur nicht mit jeder Iteration, sondern erst nach 100 Versuchen herabgesetzt wird. In Abbildung 5.12 ist der Verlauf der Log-Likelihood für verschiedene Kühlschemata wiedergegeben. Zur Beschleunigung des Algorithmus wurde die Bewertungsfunktion zu Beginn anhand einer kleinen, zufälligen Stichprobe berechnet, die im Verlauf der Iteration bis zum vollen Umfang der Trainingsmenge vergrößert wird.

Die vier dargestellten Zeitreihen zeigen deutlich die Abhängigkeit des erzielten Ergebnisses vom Kühlschema. Bei zu groß gewählter Temperatur (unterste Kurve) wird zu Beginn viel Rechenzeit ohne klare Verbesserung der Zielfunktion verschwendet. Dagegen führt der Extremfall einer konstanten Temperatur von nahezu null, bei der nur bessere Lösungen akzeptiert werden, dazu, dass das Verfahren frühzeitig in einem relativ schlechten Optimum verharrt. Entscheidend ist die Wahl eines geeigneten Temperaturverlaufes, der zwischen diesen beiden Extrempunkten liegt.

Ein Vergleich der Ergebnisse mit den aus dem Baum-Welch Training resultierenden zeigt deutlich die Unterlegenheit des Simulated Annealing. Alle untersuchten Kühlschemata (auch hier nicht dargestellte) führen in verleichbarer Rechenzeit auf ein deutlich schlechteres Modell. Versuche mit wesentlich längeren Berechnungen haben gezeigt, dass die Likelihood noch sehr verbessert werden kann.

Trainingsläufe mit kleineren Modellen und künstlich generierten Daten ergeben, dass das Simulated Annealing durchaus in der Lage ist, das optimale Modell zu finden. Dies geschieht jedoch immer auf Kosten einer gegenüber dem Training mit dem Baum-Welch-Algorithmus deutlich erhöhten Rechenzeit. Ein weiterer Nachteil des Simulated Annealing ist die hohe Sensitivität bezüglich des zugrunde gelegten Kühlschemas. Auf der anderen Seite ist das Verfahren sehr robust gegenüber Änderungen der Initialmodelle.

Als Fazit der Untersuchungen kann festgehalten werden, dass die einfachsten Methode der mul-

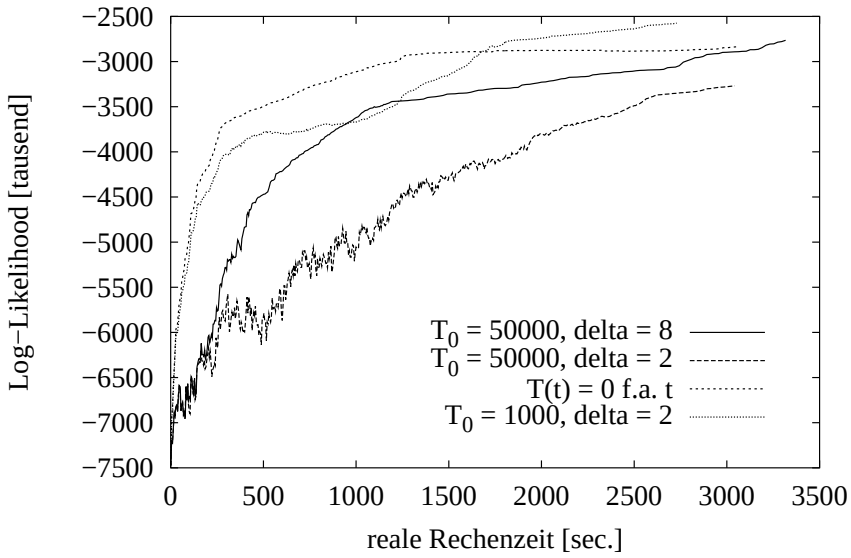

Abbildung 5.12: *Verlauf der Log-Likelihood beim Simulated Annealing unter Verwendung vier unterschiedlicher Kühlschemata (die angegebenen Parameter beziehen sich auf Gleichung* (5.5)*)*

tiplen Baum-Welch Trainingsläufe mit unterschiedlicher Initialisierung auch der beste der betrachteten Ansätze ist, mit der Problematik lokaler Extrema umzugehen.

5.4 Datenbasierte Modellwahl

Eine häufig auftauchende Frage bei modellbasierten Trainings- oder Clusterverfahren ist die nach der Wahl einer geeigneten Modelltopologie bzw. Modelldimension. Unter der Modelldimension soll hier erneut die Anzahl der freien Modellparameter verstanden werden. Bei Festlegung einer Modelltopologie bleibt bei modellbasierten Clusterverfahren damit das Problem, eine passende Anzahl von Clustern zu bestimmen. In der Literatur werden verschiedene Verfahren beschrieben, bei denen das Auffinden der passenden Topologie ein Teil des Lern- bzw. Clusteralgorithmus ist.

Bei Clusterungen werden u. a. hierarchische Verfahren [17] eingesetzt, die die Anzahl der Cluster mit jeder Iteration ändern und das Clusterergebnis anhand einer Bewertungsfunktion beurteilen.

Im Zusammenhang mit Hidden-Markov-Modellen wird in [39] ein hierarchisches Trainingsverfahren beschrieben, das auf einem Bayes-Ansatz basiert. Es startet mit einer sehr großen und speziell auf die Trainingsdaten zugeschnittenen Modelltopologie. In jeder Iteration wird die Modelldimension durch Zusammenlegung geeigneter Zustände reduziert, womit eine Verallgemeinerung des Modells einhergeht. Dort wird ebenfalls eine Bewertungsfunktion definiert, die be-

stimmt, an welcher Stelle der Algorithmus abbricht. Somit liefert das Verfahren neben den Modellparametern auch die für die Trainingsdaten optimale Modelltopologie. Zur Durchführung dieser Methode müssen für alle Modellparameter a priori Wahrscheinlichkeitsverteilungen angenommen werden, anhand derer die Likelihood eines vorgegebenen Modells berechnet werden kann. Das Verfahren maximiert die a posteriori Wahrscheinlichkeitsdichte, also die Wahrscheinlichkeit des Modells bei Vorhandensein der Trainingsdaten.

In den beiden nachfolgenden Abschnitten werden jedoch zwei alternative Lösungswege untersucht, die sich durch ihre Einfachheit in der Berechnung bzw. Durchführung auszeichnen. Mit diesen Methoden werden anschließend die Modelldimension und verschiedene Modelltopologien untersucht.

5.4.1 Bayes Information Criterion (BIC)

Das „Bayes Information Criterion“ (BIC) (auch „Schwarz Information Criterion“ genannt) wurde in [34] zum ersten Mal untersucht. Dort wird gezeigt, dass mit BIC der sogenannte Bayes-Faktor approximiert werden kann, ohne dass a priori Wahrscheinlichkeitsdichten der Modellparameter festgelegt werden müssen. Ein ausführlicher Artikel über Bayes-Faktoren ist mit [19] gegeben.

Die Verwendung des BIC ist ein wichtiges Werkzeug, um verschiedene Modellbildungen von Daten zu quantifizieren, um somit das wahrscheinlichste Modell zur Abbildung einer gegebenen Datenmenge zu bestimmen. In diesem und im nachfolgenden Abschnitt wird angenommen, dass *MKL* verschiedene Modellklassen betrachtet werden. Die Modellklasse wird mit i, wobei $1 \leq i \leq MKL$, und die Gesamtheit der Parameter der Klasse i mit Λ_i bezeichnet.

Die Entscheidung für eine bestimmte Modellklasse i kann nicht allein anhand der Wahrscheinlichkeitsdichte $p(O|\Lambda_i)$ getroffen werden, da hierbei die a priori Wahrscheinlichkeiten der unterschiedlichen Modellklassen gänzlich unberücksichtigt blieben. Sinnvoller wäre es dagegen, als beste Modellklasse jene zu wählen, die die a posteriori Modellwahrscheinlichkeit $pr(i|O)$ maximiert. Die folgenden Überlegungen zeigen jedoch, dass dies in der Praxis schwer durchführbar ist, und in [31] wird demonstriert, dass die Modellwahl anhand von BIC eine sinnvolle Alternative darstellen kann.

Für $p(\Lambda_i, i|O)$ gilt nach der Bayes-Formel

$$p(\Lambda_i, i|O) = \frac{pr(i)\, p(\Lambda_i|i)\, p(O|\Lambda_i, i)}{p(O)}. \tag{5.6}$$

Die Integration über Λ_i ergibt die gesuchte Größe:

$$pr(i|O) = p(O)^{-1} pr(i) \int p(\Lambda_i|i)\, p(O|\Lambda_i, i)\, d\Lambda_i. \tag{5.7}$$

Bei der Bestimmung der optimalen Modellklasse für das HMBM anhand dieser Formel ergeben sich zwei Probleme:

1. Das Integral kann in der Regel nicht analytisch berechnet werden. Auch eine numerische Integration kann gerade bei größeren Datenmengen nur schwer durchführbar sein.

2. Die in der Formel auftauchenden Größen $pr(i)$ (a priori Wahrscheinlichkeit der Modellklasse i) und $p(\Lambda_i|i)$ (Parameter-Prior des i-ten Modells) sind für die betrachteten Hidden-Markov-Modelle in der Regel gar nicht bekannt.

Alternativ zur Maximierung von (5.7) werden in der Literatur (z. B. in [7]) verschiedene Approximationen des obigen Ausdrucks vorgeschlagen, von denen das „Bayes Information Criterion“ bezüglich seiner Anwendbarkeit auf die Modellwahl im HMBM untersucht werden soll.

Nach diesem Kriterium wird das Modell i als wahrscheinlichstes erachtet, das

$$BIC := \log(p(O|\Lambda_i)) - \frac{1}{2} d_i \log(T), \tag{5.8}$$

maximiert [34], wobei d_i die Modelldimension bzw. die Anzahl der freien Modellparameter und T die Größe der Stichprobe bezeichnen. Im Fall einer Sequenz ist T die Länge der Sequenz, im Fall mehrere Sequenzen die Gesamtlänge aller Sequenzen. Anschaulich kann der zweite Term in (5.8) so erklärt werden, dass er einen Strafterm für eine Vergrößerung der Modelldimension darstellt. Bei zwei Modellen mit gleicher Wahrscheinlichkeitsdichte $p(O|\cdot)$ wird nach dem BIC immer das Modell mit der geringeren Parameteranzahl bevorzugt.

Nachfolgend wird das BIC zur Bestimmung der Zustandsanzahl, der Anzahl der Komponentenmodelle im MIX-HMM-Algorithmus und zur Untersuchung der Modelltopologie eingesetzt.

5.4.2 Monte-Carlo-Cross-Validierung (MCCV)

Das Verfahren der Monte-Carlo-Cross-Validierung (MCCV) wurde in [35] entwickelt und soll dazu dienen, unter verschiedenen zur Auswahl stehenden Modelltopologien oder Modelldimensionen jenes Modell zu bestimmen, das eine vorgegebene Datenmenge optimal beschreibt.

Zur Durchführung der MCCV werden die vorgegebenen Daten in eine Test- und eine Trainingsmenge unterteilt:

$$O = \{O_{Test}, O_{Train}\}.$$

In [37] wird berichtet, dass sich ein robustes Verfahren ergibt, wenn die beiden Sequenzmengen in etwa gleich groß gewählt werden. Mit W_{Test} sei die Summe der Sequenzgewichte von O_{Test} bezeichnet. Die zu vergleichenden Modellklassen $i,\ 1 \leq i \leq MKL$ werden jeweils mit der Trainingsmenge trainiert und für jedes trainierte Modell Λ_i wird die Log-Likelihood pro Einheitssequenz

$$\log(\bar{L}_{O_{Test}}(\Lambda_i)) := \frac{1}{W_{Test}} \log(L_{O_{Test}}(\Lambda_i)) = \frac{1}{W_{Test}} \sum_{O^k \in O_{Test}} \log(p(O^k|\Lambda_i)) \tag{5.9}$$

bezüglich der Testmenge berechnet.

Als Optimum wird jenes Modell gewählt, das $\log(\bar{L}_{O_{Test}}(\Lambda_i))$ maximiert. Bei der Berechnung der Likelihood bezüglich einer Testmenge, die sich von der Trainingsmenge unterscheidet, ist zu erwarten, dass die Likelihood nicht streng monoton mit der Modelldimension wächst. Oberhalb

einer bestimmter Dimension wird es zu einem Abfall der Likelihood kommen, deren Ursache in der verstärkten Spezialisierung auf die Trainingsdaten mit zunehmender Modelldimension liegt. Dies ist die entscheidende Idee bei der MCCV.

Da das Baum-Welch Training in der Regel kein globales Optimum findet, muss das oben beschriebene Verfahren unter Zugrundelegung verschiedener Trainingsmengen mehrfach durchgeführt und die Likelihood (5.9) entsprechend gemittelt werden. Meist stehen jedoch nicht genügend Trainings- bzw. Testdaten zur Verfügung. Daher wird in [36] folgendes Vorgehen vorgeschlagen: Vor jedem Trainingsdurchlauf wird die Menge der zur Verfügung stehenden Daten zufällig in eine Trainings- und eine Testmenge unterteilt. [2] Mit jedem neuen Training ändert sich folglich die Zusammensetzung dieser beiden Mengen.

Ob dieses Verfahren geeignet ist, auch in der HMM-Modellierung von Bauspardaten die optimale Modelldimension zu bestimmen, wird im Abschnitt (5.4.3) untersucht werden.

5.4.3 Modelldimension

Bestimmung der Zustandsanzahl
Ein erster Test von BIC in der HMM-Anwendung wird auf einer Menge HMM-generierter Sequenzen durchgeführt. Zu diesem Zweck werden 5000 Sequenzen der Länge $T = 20$ mit einem HMM, bestehend aus sechs Zuständen mit willkürlich festgelegten Parametern, generiert. Anhand des Kriteriums soll ermittelt werden, wie viele Zustände das Generatormodell zur Erzeugung der Sequenzen verwendet hat.

Hierzu werden die generierten Sequenzen eingesetzt, um zehn verschiedene Modelle mit einer Zustandsanzahl von eins bis zehn in getrennten Baum-Welch-Läufen zu trainieren. Die Parameter der dabei verwendeten Initialmodelle werden zufällig festgesetzt. Im nächsten Schritt wird für jede der generierten Sequenzen registriert, welches dieser trainierten Modelle gemäß dem Kriterium in Gleichung (5.8) das wahrscheinlichste Modell ist.

Im Ergebnis kann festgestellt werden, dass die korrekte Zustandsanzahl mit dem BIC deutlich unterschätzt wird. Statt der erwarteten sechs Zustände werden für die meisten der Sequenzen ein oder zwei Zustände als optimale Anzahl ermittelt. Das bedeutet, dass der Strafterm für eine Dimensionserhöhung in Gleichung (5.8) eine zu starke Gewichtung hat. Hier ist anzumerken, dass BIC eine gute Approximation des Bayes-Faktors für sehr lange Zeitreihen ist. Die vorliegenden Datensätze sind offenbar zu kurz. Alternativ zum Kriterium (5.8) soll nachfolgend untersucht werden, ob eine bessere Modellwahl mit einer verringerten Gewichtung des Straftermes erzielt werden kann. Zu diesem Zweck sollen Untersuchungen mit folgendem alternativen Kriterium durchgeführt werden:

$$BIC_g := \log(p(O|\lambda_i)) - g\, d_i \log(T). \tag{5.10}$$

Wird g hier auf null gesetzt, so geht BIC_g in die gewöhnliche Log-Likelihood über. Als weitere Alternative soll das „Akaike Information Criterion“ (AIC) untersucht werden, das wie folgt

[2] Aufgrund dieser zufälligen Unterteilung wurde das Verfahren Monte-Carlo-Cross-Validierung genannt

definiert ist [7]:

$$AIC := \log(p(O|\lambda_i)) - d_i. \tag{5.11}$$

Neben dem Einfluss des Gewichtsfaktors g soll noch die Länge der generierten Sequenzen variiert werden. Da das Generatormodell keinen Endzustand besitzt, ist dies beliebig möglich. In den Tabellen 5.5 und 5.6 sind für unterschiedliche Gewichtungen g (Gleichung (5.10)) und für das AIC unter Verwendung drei verschiedener Sequenzlängen ($T = 10, T = 20, T = 50$) wiedergegeben, bei wievielen der 5000 Sequenzen die jeweilige Zustandsanzahl optimal ist.

	g = 0.5 (BIC)			**g = 0.25**			**g = 0.1**		
N	T = 10	T = 20	T = 50	T = 10	T = 20	T = 50	T = 10	T = 20	T = 50
1	2389	545	5	679	59	0	65	2	0
2	2611	4455	4991	4266	4648	1707	1963	538	1
3	0	0	4	55	224	366	714	281	1
4	0	0	0	0	67	683	881	622	31
5	0	0	0	0	2	638	640	979	243
6	**0**	**0**	**0**	**0**	**0**	**1606**	**737**	**2578**	**4724**
>= 7	0	0	0	0	0	0	0	0	0

Tabelle 5.5: *Modellwahl mit dem Kriterium BIC_g: Ermittlung der korrekten Zustandsanzahl bei unterschiedlicher Gewichtung g und bei verschiedenen Sequenzlängen*

	g = 0.05			**g = 0 (Log-Likelihood)**			**AIC**		
N	T = 10	T = 20	T = 50	T = 10	T = 20	T = 50	T = 10	T = 20	T = 50
1	11	0	0	1	0	0	1854	155	0
2	407	43	0	29	2	0	3146	4828	1897
3	224	43	0	31	3	0	0	16	392
4	593	179	4	131	39	0	0	1	684
5	1031	686	100	538	264	30	0	0	586
6	**2734**	**4049**	**4896**	**402**	**556**	**671**	**0**	**0**	**1441**
7	0	0	0	537	430	393	0	0	0
8	0	0	0	1179	1239	1267	0	0	0
9	0	0	0	1063	1289	1396	0	0	0
10	0	0	0	1089	1178	1243	0	0	0

Tabelle 5.6: *Modellwahl mit den Kriterien BIC_g und AIC: Ermittlung der korrekten Zustandsanzahl bei unterschiedlicher Gewichtung g und bei verschiedenen Sequenzlängen*

Folgendes fällt auf:

1. Ein gewichtetes BIC mit $g = 0.05$ liefert von allen untersuchten Kriterien die besten Ergebnisse. Bei allen betrachteten Sequenzlängen liefert diese Gewichtung die korrekte Anzahl von $N = 6$.
2. Die Verwendung des Kriteriums AIC und der Log-Likelihood (g=0) führt nicht auf die korrekte Zustandsanzahl. Im Fall von AIC wird die Zustandszahl unterschätzt, während sie bei der Log-Likelihood wie zu erwarten überschätzt wird.
3. Je länger die betrachteten Sequenzen sind, desto einfacher fällt die Bestimmung der richtigen Zustandsanzahl.
4. Abgesehen von der Log-Likelihood schließen alle Kriterien die überdimensionierten Modelle ($N \geq 7$) definitiv aus. Die Ursache dieser klaren Ablehnung ist darin zu sehen, dass die trainierten Modelle mit mehr als sechs Zuständen von der Erhöhung der Dimension nur sehr wenig Gebrauch machen. Die zusätzlichen Zustände werden beim Trainieren nur sehr selten eingenommen. Der Bestrafung durch die vergrößerte Modelldimension steht somit keine deutliche Verbesserung der Abbildung der Sequenzen gegenüber.

Ausgehend von der dritten Beobachtung hat ein Versuch mit sehr langen Sequenzen (T = 500) ergeben, dass alle Kriterien abgesehen von der Log-Likelihood für jede der generierten Sequenzen (ohne Ausnahme) die korrekte Zustandsanzahl liefern.

Anzahl der Komponenten in einem HMM-Mischmodell
Nun soll die Frage untersucht werden, ob es mit der beschriebenen Methode auch möglich ist, die Anzahl der Mischkomponenten C in einem HMM-Mischmodell gegeben durch

$$p(O|\lambda, \alpha) = \sum_{j=1}^{C} \alpha_j p_j(O|\lambda_j)$$

zu erkennen. Das Vorgehen ist identisch mit dem oben erläuterten zur Bestimmung der Zustandsanzahl, nur dass diesmal das eingesetzte Generatormodell durch ein Training mit realen Bausparsequenzen des Abschlussjahrganges 1985 gewonnen wird. Es besteht aus fünf Mischkomponenten und die Alternativmodelle setzen sich zusammen aus ein bis zehn Komponenten. Als Modelldimension d bietet es sich hier an, direkt die Anzahl C der Komponenten einzusetzen. Die Likelihood in der Berechnung von BIC enthält bei jeder Sequenz nur den Beitrag der Komponente mit maximaler Likelihood; die unwahrscheinlicheren Komponenten bleiben unberücksichtigt. Versuche mit der Gesamtlikelihood haben zu keinem befriedigenden Kriterium geführt.

Dem Balkendiagramm in Abbildung 5.13 liegen 10800 generierte Sequenzen mit einer minimalen Sequenzlänge von sechs zugrunde. Kürzere Sequenzen sind unberücksichtigt, da sich bei sehr kurzen Sequenzen kein verlässliches Kriterium zur Modellwahl angeben lässt. Die Abbildung gibt wieder, wieviele der Sequenzen unter Verwendung von BIC_g, gegeben durch Gleichung (5.10) mit $g = 0.05$, bei den verschiedenen Komponentenanzahlen landen.

Es ist ersichtlich, dass die meisten Sequenzen ein Modell mit sechs Komponenten favorisieren, jedoch ist die Abgrenzung zu Modellen mit einer größeren oder kleineren Komponentenzahl nicht sehr ausgeprägt. Recht deutlich werden jedoch Modelle mit acht oder mehr Komponenten abgelehnt.

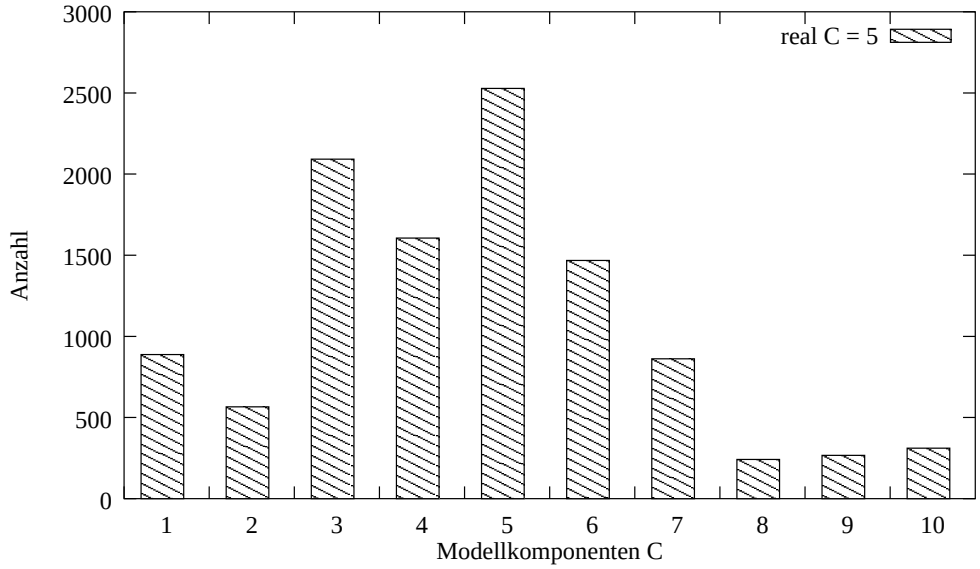

Abbildung 5.13: Modellwahl mit BIC_g bei künstlichen Daten: den generierten Sequenzen liegen fünf Komponentenmodelle zugrunde.

Alles in allem ist mit BIC_g ein brauchbares Kriterium zur Bestimmung einer geeigneten Modelldimension bei generierten Daten gegeben. Diese Modellwahl fällt naturgemäß umso leichter, je länger die betrachteten Sequenzen sind. Die Verwendung künstlicher Daten hat den Vorteil, dass die „richtige Antwort“ bekannt ist.

Komponentenanzahl bei realen Daten
Interessanter, aber wie sich herausstellen wird auch deutlich problematischer, ist die Bestimmung der Modelldimension für den Fall realer Daten. Zur Festlegung der Anzahl der Komponentenmodelle C in der Mischmodellierung gegeben durch Gleichung (5.4.3) sollen die beiden beschriebenen Kriterien eingesetzt werden. Hierzu werden die Sequenzen des Abschlussjahrganges 1985 (etwa 12 400 Sequenzen) als Trainingsmenge für Modelle unterschiedlicher Komponentenzahl benutzt. Die verwendete Topologie ist in Abbildung 4.3 auf Seite 68 dargestellt.

Zunächst soll die Fragestellung mit der MCCV-Methode beantwortet werden. Für jede zu untersuchende Modelldimension ($1 \leq C \leq 13$) wird die Sequenzmenge zufällig zu etwa gleichen Teilen in eine Trainings- und eine Testmenge unterteilt. Jeder Trainingslauf wird zehn mal pro Komponentenanzahl bei jeweils veränderter Trainings- und Testmenge wiederholt.

In Abbildung 5.14 ist die über zehn Läufe gemittelte Log-Likelihood pro Einheitssequenz für die Trainingssequenzen und für die Testsequenzen in Abhängigkeit von der Anzahl der Komponentenmodelle aufgetragen.

Die Log-Likelihood der Trainingsmenge steigt wie zu erwarten mit wachsender Komponentenanzahl nahezu monoton und ist daher nicht zur Bestimmung der Modelldimension geeignet. Interessanter ist dagegen der Verlauf der Werte bei der Testmenge. Hier ist im Bereich kleiner Anzahlen ein deutlicher Anstieg zu verzeichnen, der dann ab etwa sechs Komponenten stagniert

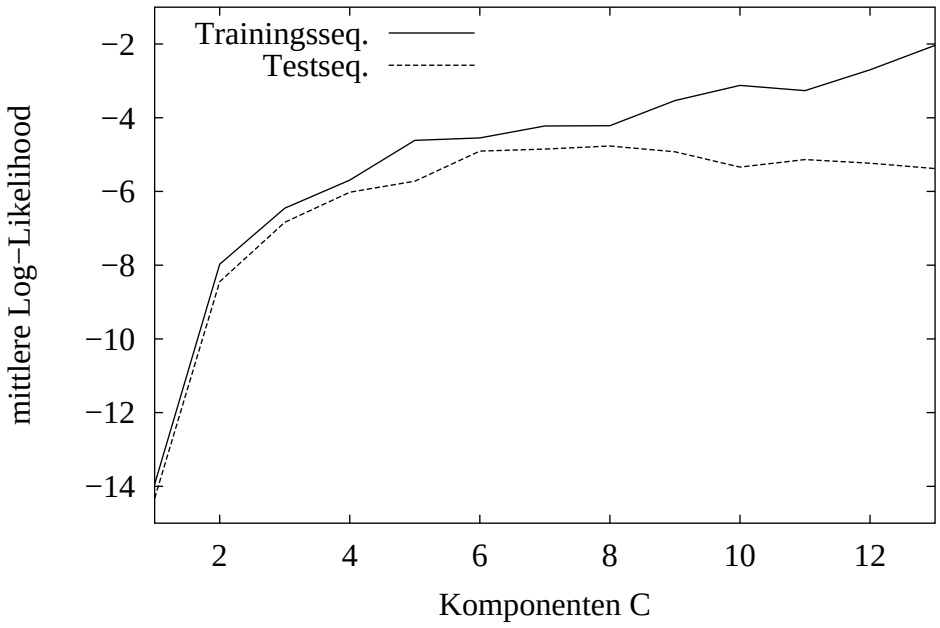

Abbildung 5.14: MCCV zur Bestimmung der Komponentenanzahl in einem HMM-Mischmodell: Dargestellt ist die gemittelte Log-Likelihood pro Einheitssequenz der Trainingssequenzen und der Testsequenzen.

und bei weiter wachsender Komponentenanzahl erwartungsgemäß leicht abfällt. Gemäß diesem Kriterium ist eine Dimension von sechs oder sieben Mischkomponenten als geeignet zu betrachten. Die größer dimensionierten Modelle liefern zwar nur unwesentlich schlechtere Log-Likelihood-Werte in der Testmenge, dennoch sind die kleineren Modelle zu bevorzugen, da sie bei geringerer Parameteranzahl eine gute Abbildungsqualität gewährleisten.

Mit dem Kriterium BIC_g werden ebenfalls Modelle mit einer Komponentenzahl von eins bis 13 untersucht. Jedes der Modelle wird mit der vollständigen Sequenzmenge trainiert und anschließend wird bestimmt, wieviele der Trainingssequenzen die einzelnen Modellen gemäß dem BIC_g-Kriterium auf sich vereinen können. Abbildung 5.15 zeigt das Ergebnis bei einer Gewichtung von $g = 0.05$, da sich diese bei den künstlichen Sequenzen als geeignet erwiesen hat.

Es zeigt sich, dass eine Komponentenanzahl von zwei bzw. acht oder neun mit diesem Kriterium gerechtfertigt werden kann. Es muss aber betont werden, dass diesen Ergebnissen eine bestimmte Gewichtung g (s. Gleichung (5.10)) zugrunde liegt. Bei einer anderen Gewichtung ergeben sich abweichende Zahlen.

Die Resultate ergeben, dass die datenbasierte Modellwahl bei realen Bauspardaten wesentlich schwieriger ist als bei künstlich generierten Daten. Die Ursache hierfür dürfte in den beiden folgenden Punkten liegen:

1. Den realen Bauspardaten liegt keine klare Trennung in verschiedene Komponenten zu-

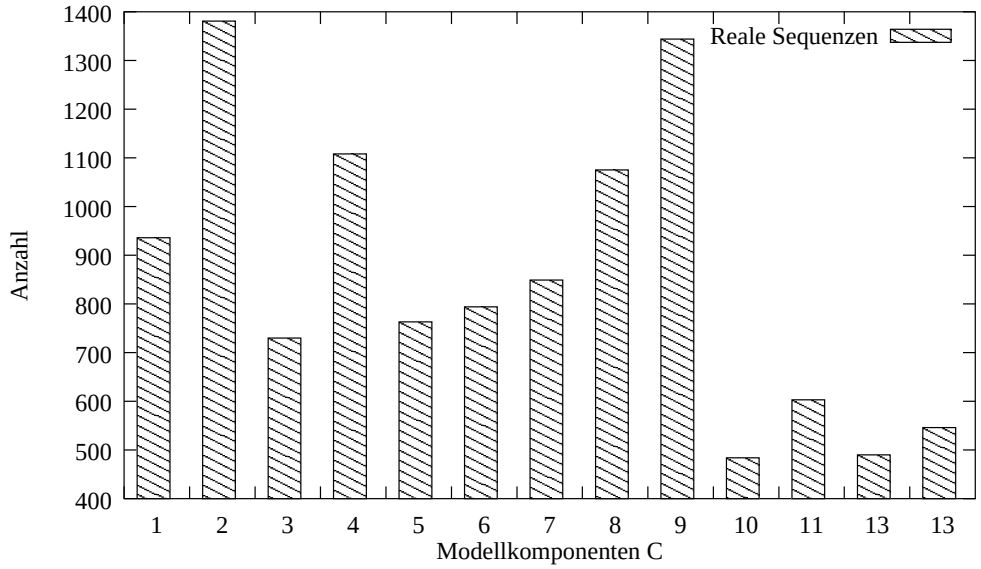

Abbildung 5.15: *Wahl der Komponentenzahl im Mischmodell mit BIC_g bei bei realen Daten*

grunde. Die Übergänge sind fließend und damit erschwert sich die Suche nach einer optimalen Anzahl der Submodelle.

2. Das Kriterium BIC ist gerade bei sehr langen Sequenzen ein guter Indikator. Je kürzer die Sequenzen sind, desto ungeeigneter wird dieses Kriterium. Bei den Bausparsequenzen gibt es gerade durch einen signifikanten Anteil von Schnellsparern und Kündigern sehr viel kurze Sequenzen.

Zusammenfassend kann gesagt werden, dass die Verwendung des BIC_g und der MCCV-Methode bei der gegebenen Modelltopologie ein Mischmodell bestehend aus sechs bis neun Komponenten anzeigen. Die Vorgabe von nur zwei Modellkomponten, ist gemäß der MCCV-Methode nicht hinreichend, da die mittlere Log-Likelihood sowohl bezüglich der Trainings- als auch der Testsequenzen zu niedrig ist.

5.4.4 Varianten in der Modelltopologie

In diesem Abschnitt sollen verschiedene Modelltopologien hinsichtlich ihrer Eignung zur Abbildung realer Bausparsequenzen untersucht werden. Dabei werden ausschließlich die Bereiche der Sparphase variiert, da sich die Topologie der Darlehensphase auf naheliegende Weise durch die strikten Rahmenbedingungen des Bausparens bezüglich der Tilgung ergibt. In den nachfolgend betrachteten Modellen wird für die Darlehensphase eine Topologie gemäß Abbildung 4.3, S. 68 verwendet. Die folgenden Modelltopologien, charakterisiert durch die erlaubten Zustandswechsel, mit jeweils fünf Sparzuständen werden miteinander verglichen:

FC (fully connected): alle Übergänge zwischen Sparzuständen sind möglich

LR (links rechts): Selbstübergänge und Übergänge in alle Nachfolgezustände sind erlaubt (s. Abbildung 5.16)

SLR (strikt links rechts): Selbstübergänge und Übergänge in den jeweils unmittelbaren Nachfolgezustand sind zulässig

BSPK (bauspar): speziell auf die Gegenheiten der Sparphase im Bausparwesen anpasste Modelltopologie gemäß Abbildung 5.17.

Die Initialwahrscheinlichkeit der Zustände ist so vorgegeben, dass der Zustandspfad in jedem Sparzustand starten kann. Lediglich im BSPK-Modell ist ein Start im letzten Zustand (Zuzahler) unterbunden, da dies aus bauspartechnischer Sicht keinen Sinn macht.

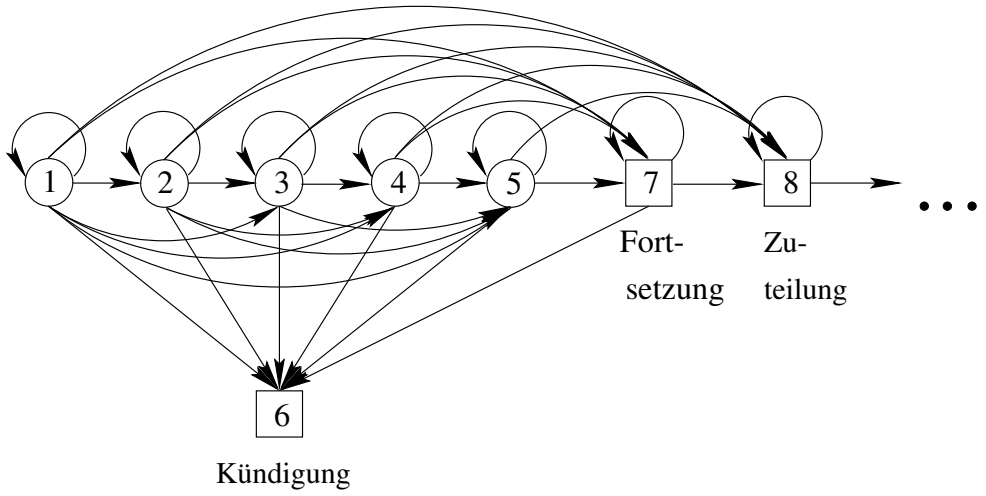

Abbildung 5.16: *„Links Rechts" Modelltopologie (LR), nur Sparphase*

Die Festlegung der möglichen Zustandswechsel in der BSPK-Topologie ist durch drei in den Daten häufig vorkommende grundlegende Verhaltensmuster während der Sparphase motiviert. Bei der Soforteinzahlung wird zu Beginn der Sparphase einmalig ein hoher Betrag (i. d. R. 40% oder 50% der Bausparsumme) eingezahlt. Der Regelsparer leistet jeden Monat die gleiche Einzahlung und der Zuzahler beendet seine Sparzahlung mit einer einmalig hohen Einzahlung, deren Höhe so gewählt ist, dass die baldige Zuteilung erreicht wird. Kombinationen dieser drei Grundmuster kommen auch häufig vor.

Die vier Modelltopologien sollen mit dem BIC_g-Kriterium aus Abschnitt 5.4.1 miteinander verglichen werden. Dazu wird zu jeder der Topologien ein Mischmodell bestehend aus fünf Modellkomponenten mit dem MIX-HMM-Algorithmus unter Verwendung der Daten des Abschlussjahrganges 1985 trainiert. In Tabelle 5.7 sind die Resultate mit einer Gewichtung von $g = 0.05$ (Gleichung (5.10)) für die Trainingssequenzen mit einer Mindestlänge von fünf Symbolen wiedergegeben. Die Zahlen basieren auf den ebenfalls in der Tabelle angegebenen Modelldimensionen (sechs Übergangsklassen). Zum Vergleich sind in der Tabelle auch die Zahlen angegeben, die sich ohne Verwendung eines Straftermes für die Modelldimension ($g = 0$) ergeben:

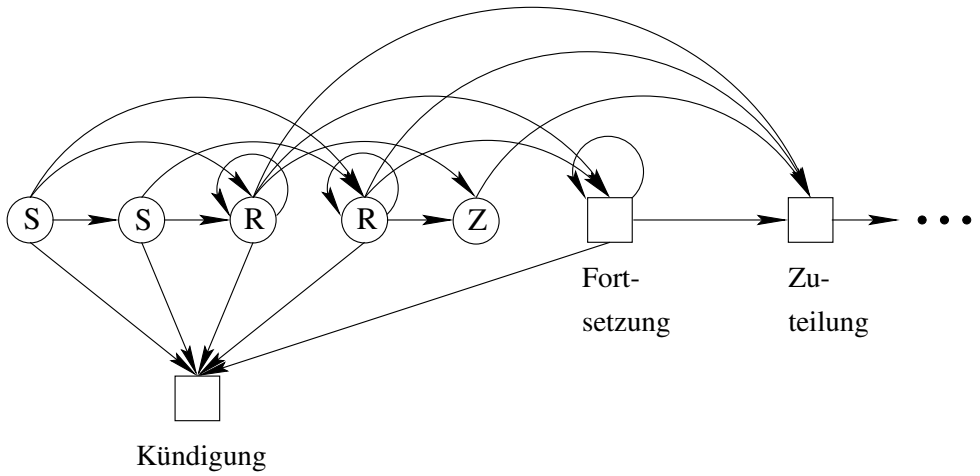

Abbildung 5.17: *Eine speziell angepasste Bausparmodelltopologie (BSPK) mit Vorgabe der drei Einzahlungsmoden „Sorforteinzahler“ (S), „Regelsparer“ (R) und „Zuzahler“ (Z)*

Modell	FC	LR	SLR	BSPK
freie Parameter d	257	196	177	135
$g = 0.05$	426	688	2191	7783
$g = 0$	5066	2041	1948	2033

Tabelle 5.7: *Topologiewahl mit dem Kriterium BIC_g:*

Die Bauspartopologie liefert von allen Modellen das beste Ergebnis, wenn als Kriterium das erprobte BIC_g mit einer Gewichtung von $g = 0.05$ eingesetzt wird. Bei Fortlassen des Straftermes ($g = 0$) erscheint dagegen die Topologie „FC“ optimal. Wie können diese Ergebnisse interpretiert werden? Offensichtlich führt die vollständig verbundene Modelltopologie zur besten Abbildung der Trainingsdaten, dies jedoch auf Kosten einer fast verdoppelten Modelldimension im Vergleich zur BSPK-Topologie. Die Topologie „FC“ ist das allgemeinste Modell und enthält die übrigen drei Topologien als Spezialfall. Somit ist es plausibel, dass sich dort die besten Likelihood-Werte ergeben. Die Modelle „BSPK“, „LR“ und „SLR“ sind bezogen auf die Likelihood nahezu gleichwertig. Da das BSPK-Modell diese Abbildungsqualität mit der kleinsten Anzahl freier Parameter erreicht, schneidet diese Topologie bei Berücksichtigung eines Straftermes, unabhängig von der dabei verwendeten Gewichtung, am besten ab.

Die weitere Analyse der Modelltopologien anhand der Likelihood bezogen auf eine Testmenge und anhand der Eigenschaften beim Generieren von Sequenzen führt zu keinem klaren Votum für oder gegen eine bestimmte Modelltopologie.

Somit gilt für die Anwendung von Hidden-Markov-Modellen in der Bausparmodellierung, was auch aus anderen Anwendungsgebieten bekannt ist [32]: Es ist sinnvoll, bei der Wahl der Modelltopologie dataillierte Kenntnisse über das zu modellierende System einzubringen.

5.4.5 Ausgabefunktionen

Im Abschnitt 4.4.4 wurde erläutert, dass die Normaldichte und die bei null gestutzte Normaldichte als Ausgabefunktionen für die Zustände im HMBM in Betracht kommen. Ein Vergleich dieser beiden Ausgabefunktionen durch die Berechnung einer Abstandsfunktion, die Betragsdifferenzen wichtiger Kollektivzeitreihen summiert, jeweils angewendet auf künstlich verlängerte Sequenzen und die entsprechenden realen Sequenzen zeigt eine geringfügig bessere Abbildungsqualität für den Fall der herkömmlichen Dichtefunktion. Die Unterschiede sind jedoch so gering, dass sie kein auschlaggebendes Kriterium für oder gegen die Verwendung einer bestimmten Ausgabefunktion sein können. Aus Sicht der Prognosegüte sind beide Ausgabefunktionen für die Modellierung der Bauspardaten geeignet.

Als ein Beleg kann die Abbildung 5.18 dienen. Sie gibt wieder, wie sich die Sparzahlungen in den Realdaten und in den generierten Daten verteilen. Dabei sind die beiden Generatormodelle auf der gleichen Datenbasis mit dem MIX-HMM Algorithmus ermittelt worden. Die verwendeten Initialmodelle unterschieden sich lediglich in den Ausgabefunktionen. Beide Modelle sind in der Lage, die Verteilung der realen Spargeldeingänge mit geringen Abweichungen zu reproduzieren.

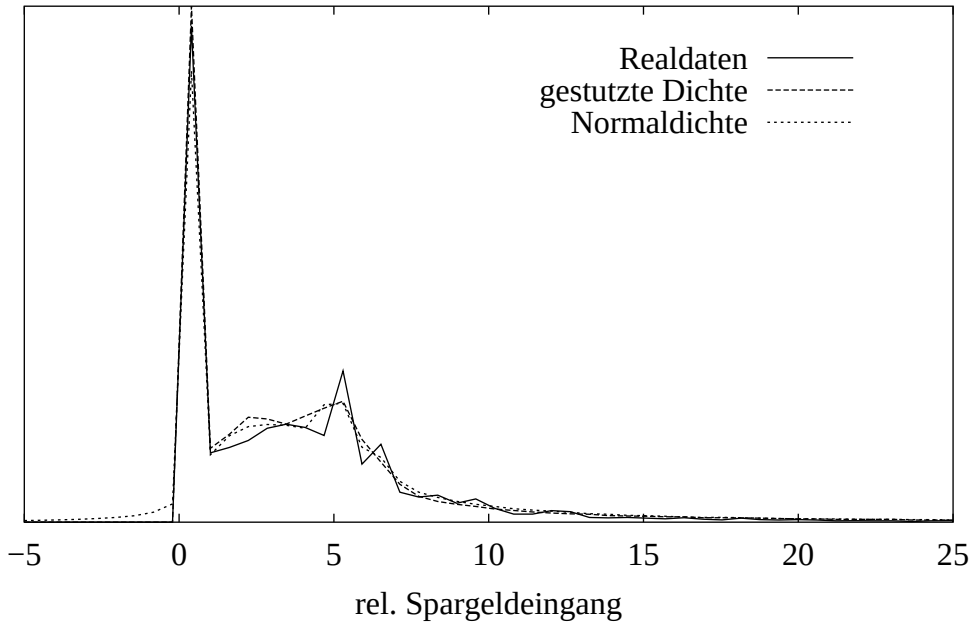

Abbildung 5.18: *Verteilung realer und generierter Spargeldeingänge unter Verwendung der gestutzten und der nicht gestutzten Gauß-Dichte als Ausgabefunktion*

In 4.4.4 wurde auch auf die Problematik der Generierung negativer Spargeldeingänge hingewiesen, die sich bei Verwendung der herkömmlichen Normalverteilung ergeben können, auch wenn sie in den Trainingsdaten nicht vorkommen. Dass der Anteil negativer Werte beim Generieren recht gering ausfällt, ist in Abbildung 5.18 zu sehen. Kann dieser Anteil, wie in 4.4.4 vorgeschla-

gen, durch den Einsatz eines speziellen Nullzustandes verringert werden? Aufschluss hierüber gibt die Abbildung 5.19.

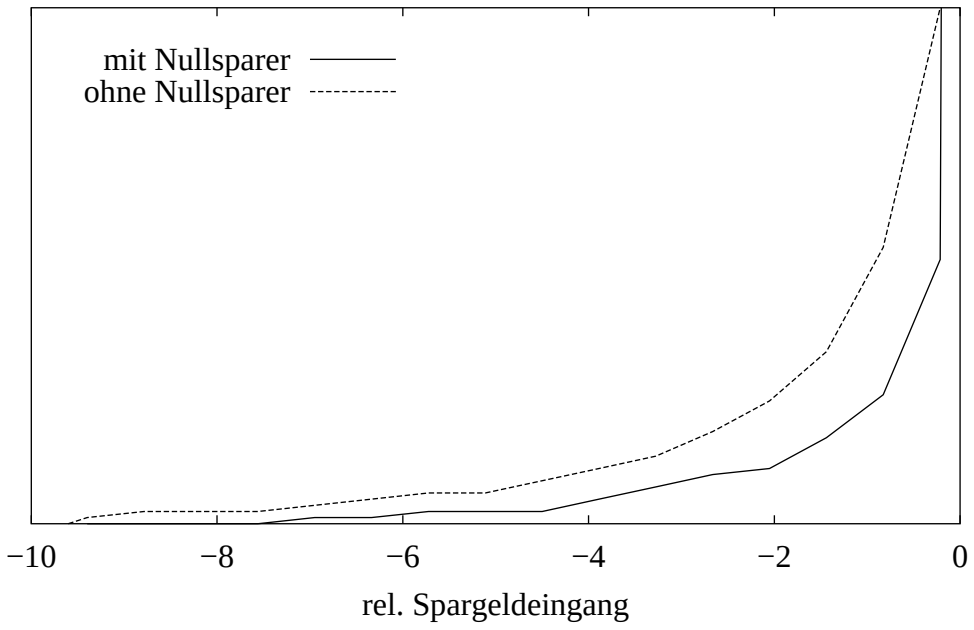

Abbildung 5.19: *Verteilung negativer generierter Spargeldeingänge mit und ohne Nullzustand (jeweils Gauß-Dichte als Ausgabefunktion)*

Die Verwendung des Nullspar-Zustandes führt zu deutlich weniger negativen Spargeldeingängen gegenüber dem Versuch ohne Nullsparer. Insgesamt ergibt sich im vorliegenden Fall mit Verwendung des Nullzustandes eine Wahrscheinlichkeit für die Generierung negativer Spargeldeingänge von etwa zwei Prozent und ohne Einsatz des Nullzustandes von etwa vier Prozent.

Die Einführung des Nullzustandes hat also den gewünschten Effekt zur Folge. Da sich in vielen Modellen auch ohne explizite Vorgabe eines Nullzustandes nach dem Training ein dem Nullzustand sehr ähnlicher Zustand herausbildet, sind die sonstigen Unterschiede in den Ergebnissen der beiden Verfahren eher gering. Für die Verwendung des Nullzustandes spricht aber ein gewisser Laufzeitvorteil, da die Berechnung der Ausgabeparameter des Zustandes entfällt.

Zusammenfassend kann die Verwendung der herkömmlichen Normaldichte in Kombination mit dem Nullzustand im HMBM als geeignete Wahl betrachtet werden, da

- die Generierung negativer Symbole mit vernachlässigbarer Wahrscheinlichkeit vorkommt,
- die Laufzeit des Trainings geringer ausfällt als bei der gestutzten Verteilung,
- keine Probleme mit numerischen Instabilitäten wie im Fall der gestutzten Verteilung (siehe [23]) auftreten.

Kapitel 6

Bausparkollektivsimulationen mit dem HMBM

Die Simulation eines realen Bausparkollektivs mit dem HMBM besteht nach erfolgtem Training der Modelle in der Durchführung folgender Schritte:

1. Erzeugung der Bestandssequenzen
2. Bestimmung eines geeigneten Generatormodells für jede Bestandssequenz
3. Generierung der Sequenzfortsetzungen
4. Generierung neuer Sequenzen (Neugeschäft)
5. Berechnung relevanter Kollektivzeitreihen aus allen generierten Sequenzen

Die Punkte eins und zwei sind der Bestandsabbildung zuzurechnen, die im ersten Abschnitt des Kapitels erläutert wird. Daran anschließend werden Untersuchungen zur Generierung und Fortsetzung von Sequenzen vorgestellt. Das Kapitel endet mit der Durchführung und Bewertung einer vollständigen Kollektivsimulation.

6.1 Bestandsabbildung

Der Verlauf wichtiger Zeitreihen eines Bausparkollektives während eines Zeithorizontes von etwa einem bis fünf Jahren, ausgehend vom Simulationsbeginn, wird sehr stark von den Verträgen geprägt, die zum Simulationsstart bereits vorhanden sind. Diese Verträge werden als Bestand des Kollektivs bezeichnet. Somit ist eine gute Bestandserfassung sehr wesentlich für die Güte jeder Bausparsimulation.

Die Bestandserfassung setzt sich im HMBM aus den folgenden beiden Punkten zusammen:

Erzeugung von Bestandssequenzen: Jeder zum Simulationsbeginn bestehende Vertrag (nur Verträge, die das Vertragsende noch nicht erreicht haben) wird durch genau eine Sequenz abgebildet. Details hierzu finden sich im Abschnitt 4.3.2.

Klassifizierung: Die Klassifizierung bzw. Zuordnung einer Bestandssequenz zu einem Modell besteht darin festzulegen, mit welchem Modell oder auch mit welchen Modellen, die Sequenz verlängert werden soll.

Zur Zuordnung der Bestandssequenzen wurden im Abschnitt 2.3.4 die Klassifikatoren MD und MAW vorgestellt. Zusätzlich zur Wahl eines dieser beiden Klassifikatoren sind die folgenden drei Vorgehensweisen bei der Zuordnung einer Sequenz möglich:

1. Zuordnung der Sequenz zum jeweils besten Modell (Best)
2. Zufällige Wahl des Generatormodells gemäß der Verteilungsfunktion bei gegebenem Klassifikator (Distr)
3. Aufteilung der Sequenz auf die zur Auswahl stehenden Modelle entsprechend der Verteilungsfunktion. Die Gewichtung der Sequenzen ermöglicht eine genaue Aufteilung entsprechend der Modellwahrscheinlichkeit $pr(\lambda_i|O)$ auf die Modelle (All)

In Kombination mit den beiden Klassifikatoren ergeben sich hieraus sechs Möglichkeiten der Zuordnung. Inwieweit sich diese verschiedenen Zuordnungen auf die Simulation des Kollektivs auswirken, soll durch einen Vergleich realer und generierter Sequenzen untersucht werden. Hierzu werden sechs, mit dem Abschlussjahrgang 1985 unter Verwendung des MIX-HMM-Algorithmus trainierte Modelle eingesetzt. Jede der Trainingssequenzen wird zu einem festen Zeitpunkt (1992) gekürzt und basierend auf dieser kurzen Anfangssequenz mit der jeweiligen Zuordnungsmethode einem oder mehreren der trainierten Modelle zugeordnet. Im nächsten Schritt werden die fehlenden Symbole jeder Sequenz unter Verwendung des assoziierten Generatormodells erzeugt. Mit dem Kollektivgenerator (s. Abschnitt 6.4) werden daraus Kollektivzeitreihen generiert.

In der Tabelle 6.1 sind die mittleren prozentualen Abweichungen wichtiger generierter Kollektivzeitreihen zu den realen Daten dargestellt. Die Mittelung erstreckt sich bei allen Versuchen über die Jahre 1993 bis einschließlich 1998. Bei jeder Kollektivgröße ist die kleinste Abweichung jeweils fett markiert.

Aus diesen Zahlen und aus der Analyse der Zeitreihen kann bezüglich der Zuordnung in der Bestandsabbildung Folgendes festgestellt werden:

- Die Zuordnung mit dem Klassifikator MAW liefert bessere Ergebnisse als die Verwendung des MD. Somit ist es sinnvoll, die aus dem Training gewonnenen a priori Wahrscheinlichkeiten der Modelle bei der Zuordnung zu berücksichtigen.
- Die Modellwahl gemäß „MAW Distr“ führt bei fünf der untersuchten Zeitreihen zur besten Übereinstimmung, während „MAW Best„ und „MAW All“ bei jeweils drei Zeitreihen vorzuziehen sind. Jedoch ergeben sich bei den aus Sicht einer Kollektivsimulation besonders wichtigen Zeitreihen „Zuteilung“ und „Spargeldeingang“ die besten Ergebnisse mit dem Vorgehen „MAW all“.
- Der Grad der Übereinstimmung ist bei den betrachteten Kollektivzeitreihen sehr unterschiedlich. Bei der Gesamt-Bausparsumme ergibt sich bei allen Methoden eine beachtlich

	MAW Distr	MAW Best	MAW All	MD Distr	MD Best	MD All
BS	**0.808**	0.835	0.905	1.028	1.224	0.913
Z_BS	**3.304**	3.824	3.467	3.834	3.774	3.681
NZ_BS	**5.319**	6.846	5.689	7.276	8.081	6.035
SPE	13.134	15.365	**10.847**	17.963	13.894	13.149
FortA	12.803	14.77	11.088	13.673	**8.835**	10.491
FortE	**21.809**	30.486	30.047	33.933	35.603	29.332
FortBst	**3.432**	5.229	4.519	5.312	4.85	4.506
Zut	19.877	18.72	**17.295**	22.248	22.998	17.497
DV	158.572	**131.974**	139.535	148.354	165.264	149.354
DN	39.765	**31.27**	37.228	43.696	41.809	41.901
Tilg	9.888	10.262	**8.38**	11.811	11.088	10.7
KBS	20.86	**17.997**	21.66	19.929	29.329	22.685

Tabelle 6.1: *Prozentuale Abweichungen der Kollektivzeitreihen mit den Abkürzungen: BS = Baussparsumme, Z_BS = zugeteilte Bausparsumme, NZ_BS = nicht zugeteilte Bausparsumme, SPE = Spargeldeingang, FortA = Fortsetzungsanfang, FortE = Fortsetzungsende, FortBst = Fortsetzer-Bestand, Zut = Zuteilungen, DV = Darlehensverzicht, DN = Darlehensnahme, Tilg = Tilgungen, KBS = gekündigte Bausparsumme*

gute Übereinstimmung, während sich bei den Darlehensverzichten und den Darlehensnahmen erhebliche Abweichungen einstellen. Eine genauere Analyse und Bewertung dieser Feststellung wird im Abschnitt 6.3 bei der Durchführung einer vollständigen Kollektivsimulation vorgenommen.

- Generell führt eine Sequenzverlängerung mit dem jeweils besten Modell im Vergleich zu den beiden anderen Methoden zu einem weniger glatten Verlauf der Zeitreihen.

Das abschließende Fazit dieser Untersuchung lautet, dass die Verwendung von a priori Modellwahrscheinlichkeiten mittels des MAW Klassifikators sinnvoll ist. Die Wahl zwischen den Methoden „Distr“, „Best“ und „All“ fällt nicht so eindeutig aus, leichte Vorteile ergeben sich bei der zufälligen Modellwahl unter Berücksichtigung der a posteriori Modellwahrscheinlichkeiten („Distr“). Die Aufteilung der Sequenzen auf die zur Verfügung stehenden Modelle liefert ebenfalls gute Ergebnisse, ist jedoch mit einem erheblichen Datenzuwachs verbunden, da jede Anfangssequenz mehrfach kopiert werden muss.

6.2 Generieren und Fortsetzen von Sequenzen

Bei verschiedenen Untersuchungen in diesem und im vorhergehenden Kapitel hat die Generierung künstlicher Sequenzen und die Fortsetzung vorhandener Teilsequenzen eine wesentliche Rolle gespielt. Dieser Aspekt des HMBM soll in diesem Abschnitt genauer beleuchtet werden.

Nach einer Beschreibung der prinzipiellen Möglichkeiten der Sequenzgenerierung und Sequenzverlängerung unter Verwendung eines Zufallszahlengenerators, werden diese anhand der Verlängerung abgeschnittener Sequenzen miteinander verglichen. Im Anschluss daran wird die

Stabilität der Sequenzgenerierung unter dem Einfluss verschiedener Initialisierungen des Zufallszahlengenerators untersucht.

Hidden-Markov-Modell als Sequenzgenerator
Ein trainiertes Hidden-Markov-Modell läßt sich sehr einfach als Generator künstlicher Sequenzen einsetzen. Hierzu ist in jedem Zeitschritt die Durchführung der beiden folgenden Schritte notwendig:

1. Festlegung des Generatorzustandes
2. Erzeugung einer Ausgabe unter Verwendung der Ausgabefunktion des Generatorzustandes

Diese beiden Schritte werden solange wiederholt, bis entweder eine vorgegebene Sequenzlänge erreicht ist oder die Sequenz nicht weiter verlängert werden kann, da ein Endzustand erreicht ist, der ein Sequenzabschlusssymbol (z. B. Kündigung oder Darlehensverzicht) generiert.

Die Wahl der Generatorzustände geschieht unter Verwendung eines Zufallszahlengenerators, mit dem eine Folge von auf dem Intervall $[0,1]$ gleichmäßig verteilten Zufallszahlen erzeugt wird. Abgesehen vom Startzustand, der eine unten beschriebene Sonderbehandlung erfordert, wird zu jedem Zeitpunkt entsprechend der Übergangswahrscheinlichkeiten des aktuellen Zustandes der neue Generatorzustand „ausgewürfelt“. Auf diese Art und Weise ist gewährleistet, dass die generierte Sequenz auf einem erlaubten Zustandspfad basiert. Die Generierung sehr vieler Sequenzen mit dieser Methode approximiert die Verteilung der zugrunde liegenden Markov-Kette.

Es hat sich herausgestellt, dass eine vereinfachte Sequenzgenerierung, bei der als neuer Generatorzustand jeweils der Zustand mit der größten Übergangswahrscheinlichkeit vom aktuellen Zustand deterministisch gewählt wird, nicht sinnvoll ist. Hierdurch werden die potenziell möglichen Zustandspfade auf einige sehr wenige reduziert, was zur Folge hat, dass bestimmte, in den Trainingssequenzen selten auftretende Aktionen in den generierten Sequenzen gar nicht vorkommen.

Zur Festlegung des Initialzustandes muß zwischen dem Generieren des Startsymbols einer neuen Sequenz und der Verlängerung einer bestehenden Anfangssequenz unterschieden werden. Im ersten Fall ist π, die Initialwahrscheinlichkeit der Zustände maßgeblich. Mit Hilfe einer gleichverteilten Zufallszahl wird mit ihr der erste Generatorzustand bestimmt. Im Fall der Sequenzverlängerung muss zunächst geklärt werden, in welchem Zustand sich das Modell bei der Ausgabe des letzten Symbols befunden hat, um dann im zweiten Schritt unter Verwendung der entsprechenden Übergangswahrscheinlichkeiten den ersten Generatorzustand festzulegen. Die folgenden drei Möglichkeiten bzgl. der Wahl des letzten Zustandes sollen hier untersucht werden:

VIT: Einsatz des Viterbi-Zustandes: Die Übergangswahrscheinlichkeiten des Endzustandes im Viterbi-Pfad (s. Abschnitt 2.3.2) unter Zugrundelegung der Anfangssequenz $O_{(t)}$ bestimmen per Zufallszahl den Ausgangszustand für die generierte Sequenz.

DISTR: Zufällige Wahl eines Zustandes gemäß der Zustandsverteilung bei gegebener Anfangssequenz: $pr(q_i|O_{(t)}),\ 1 \leq i \leq N$. Diese Größen lassen sich mit dem Forward-Algorithmus

berechnen, denn unter Verwendung der Forward Variablen $\alpha_t(i)$ gilt:

$$pr(q_i|O_{(t)}, \lambda) = \frac{\alpha_t(i)}{p(O_{(t)}|\lambda)}.$$

BEST: Wahl des besten Zustandes:

$$q_{init} = \arg\max_{q_i} pr(q_i|O_{(t)}, \lambda).$$

Eine weitere Alternative beim Generieren von Sequenzen besteht darin, jedes generierte Symbol als Teil der realen Sequenz aufzufassen und entsprechend der oben geschilderten Vorgehensweise in die Wahl des Generatorzustandes einfließen zu lassen. In der folgenden Darstellung der Untersuchungsergebnisse wird dieses Vorgehen mit **„MI“** (multiple Initialisierung) bezeichnet, während Versuche mit einmaliger Festlegung des Initialzustandes die Notation **„EI“** (einmalige Initialisierung) erhalten.

Zur Generierung der Symbole wird ebenfalls ein Zufallszahlengenerator eingesetzt, der Zahlen generieren muss, die entsprechend der Ausgabefunktion des jeweiligen Generatorzustandes verteilt sind. Dies ist im Fall der Normalverteilung durch eine einfache Transformation gleichmäßig verteilter Zufallszahlen möglich [10]. Eine Methode zur Erzeugung von Zufallszahlen mit zugrunde liegender gestutzter Normalverteilung ist in [23] zu finden. Die Erzeugung von Zufallszahlen aus einer Überlagerung mehrerer Ausgabefunktionen kann einfach erreicht werden, indem im ersten Schritt die Ausgabenkomponente zufällig festgelegt und im zweiten Schritt hierzu das Symbol wie beschrieben erzeugt wird.

	MI_BEST	MI_DISTR	MI_VIT	EI_BEST	EI_DISTR	EI_VIT
BS	0.877	0.981	1.053	1.278	**0.808**	0.862
Z_BS	4.01	3.417	3.857	**3.068**	3.304	3.346
NZ_BS	7.541	**4.477**	7.637	6.539	5.319	5.377
SPE	12.467	**10.204**	12.125	15.622	13.134	11.235
FortA	14.236	21.082	**11.617**	12.925	12.803	19.741
FortE	32.646	38.857	29.719	32.312	**21.809**	40.684
FortBst	6.141	6.218	5.081	5.776	**3.432**	7.168
Zut	17.688	**15.39**	18.76	16.734	19.877	17.783
DV	131.634	146.229	144.168	124.304	158.572	**118.483**
DN	31.83	**29.478**	34.302	31.318	39.765	38.132
Tilg	9.595	**8.417**	9.802	8.253	9.888	11.458
KBS	20.541	22.002	23.236	27.953	20.86	**17.728**

Tabelle 6.2: *Prozentuale Abweichungen der Kollektivzeitreihen mit den Abkürzungen: BS = Baussparsumme, Z_BS = zugeteilte Bausparsumme, NZ_BS = nicht zugeteilte Bausparsumme, SPE = Spargeldeingang, FortA = Fortsetzungsanfang, FortE = Fortsetzungsende, FortBst = Fortsetzer-Bestand, Zut = Zuteilungen, DV = Darlehensverzicht, DN = Darlehensnahme, Tilg = Tilgungen, KBS = gekündigte Bausparsumme*

Zur Untersuchung der verschiedenen Möglichkeiten bei der Sequenzverlängerung wird die gleiche Vorgehensweise wie in 6.1 bei der Analyse der Bestandsabbildung gewählt. Der Abschlussjahrgang 1985 wird ab 1992 unter Verwendung der Zuordnung „MAW Distr“ verlängert und die prozentualen Abweichungen der simulierten Kollektivzeitreihen von den realen Daten sind in der Tabelle 6.2 wiedergegeben.

Aus den dargestellten Zahlen können folgenden Schlüsse gezogen werden:

- Die Wahl des Initialzustandes mit der Methode „DISTR“ bietet gegenüber den beiden anderen Vorgehensweisen Vorteile bei vergleichbarem Rechenaufwand. Die Methode „BEST“ schneidet in diesem Versuch am schlechtesten ab.
- Die multiple Initialisierung („MI“) liefert in dem untersuchten Fall geringfügig bessere Resultate als die einfache Initialisierung („EI“). Jedoch ist der Vorteil nicht sehr ausgeprägt, so dass die Aussage nicht direkt auf andere Daten und Modelle übertragen werden kann. Die multiple Initialisierung ist mit einem erhöhten Rechenaufwand verbunden, da der Ausgangszustand zu jedem Zeitpunkt neu zu bestimmen ist.

Zusammenfassend kann festgestellt werden, dass es sowohl bei der Zuordnung einer Sequenz zu einem Komponentenmodell, als auch bei der Bestimmung des Endzustandes bei der Sequenzverlängerung sinnvoll ist, jeweils die gesamte Verteilungsfunktion einfließen zu lassen.

Schwankungen bei der Sequenzgenerierung
Es liegt in der Natur des gewählten stochastischen Modellansatzes, dass zwei Simulationsläufe mit dem HMBM mit exakt gleichen Parametern und gleichen sonstigen Rahmenbedingungen, aber unterschiedlicher Initialisierung des zur Sequenzgenerierung verwendeten Zufallszahlengenerators, im Allgemeinen zwei unterschiedliche Resultate liefern. Daher ist die Frage interessant, wie stark die zu erwartenden Schwankungen der Ergebnisse sind. Beispielhaft sei im Folgenden die generierte Kollektivzeitreihe der Spargeldeingänge betrachtet.

Eine analytische Berechnung der Varianzen scheint angesichts der sehr komplexen Struktur der mit dem Hidden-Markov-Modell verknüpften Verteilungsfunktion schwer durchführbar. Aber klar ist, dass der zentrale Grenzwertsatz [6] Anwendung finden kann, nach dem die Summe identisch verteilter, voneinander unabhängiger Zufallsgrößen normalverteilt ist. Dies trifft auf den kollektiven Spargeldeingang zu, da er sich aus den Beiträgen voneinander unabhängiger Sequenzen zusammensetzt, die Realisierungen einer identischen Verteilungsfunktion sind.

Um einen Eindruck von der Größe der auftretenden Schwankungen zu gewinnen, wird folgendes Experiment durchgeführt: Unter Verwendung eines mit dem MIX-HMM-Algorithmus trainierten Hidden-Markov-Modells, das aus sechs Komponentenmodellen besteht, werden in mehreren Durchläufen jeweils 12 000 Sequenzen generiert. Dies ist die Größenordnung einer Tarifklasse eines Abschlussjahrganges der betrachteten Bausparkasse. Nach jedem der Generationsläufe werden die Spargeldeingänge nach Zeitpunkten (Jahren) getrennt summiert. Die Abbildung 6.1 zeigt das Histogramm der summierten Spargeldeingänge des zweiten und dritten Jahres, das auf 500 Generationsläufen mit jeweils unterschiedlichem Startwert des Zufallszahlengenerators basiert.

Wie nach dem zentralen Grenzwertsatz zu erwarten, ergeben sich näherungsweise zwei Gaußdichten, mit gewissen Abweichungen, die aus der begrenzten Anzahl der Versuche und der recht

groben Unterteilung beim Erstellen des Histogramms (Intervallbreite von 250 Einheiten) resultieren.

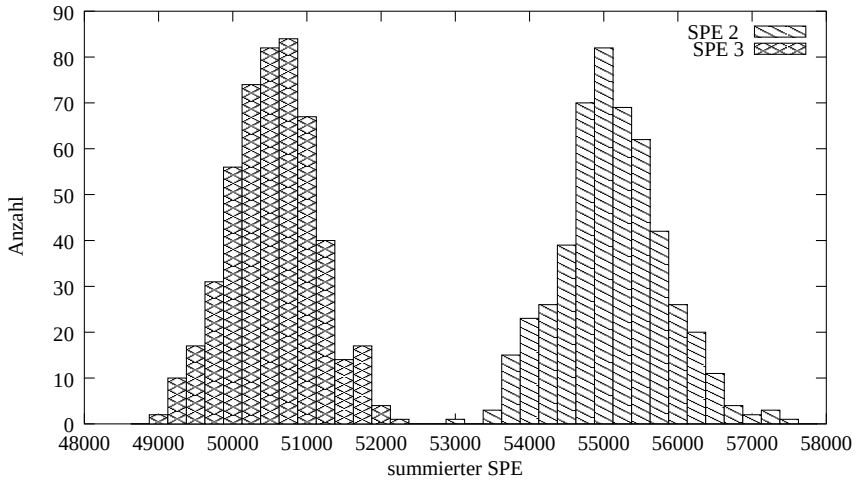

Abbildung 6.1: *Histogramm summierter SPE des zweiten und dritten Jahres aus der Generierung von* 12 000 *Sequenzen bei 500 verschiedenen Initialisierungen des Zufallszahlengenerators*

In der Tabelle 6.3 sind für die generierten summierten Spargeldeingänge der ersten neun Jahre die Mittelwerte, die Standardabweichungen und die prozentualen Standardabweichungen bezogen auf den jeweiligen Mittelwert angegeben. Schwankungen in der Größenordnung von einem bis zwei Prozent bezogen auf den Erwartungswert sind im Bereich der Bausparsimulationen gut zu akzeptierende Werte. Die Daten basieren auf den Fall der Simulation eines Neugeschäftsjahrganges. Da eine Kollektivsimulation sich aus der Überlagerung mehrerer Jahrgänge zusammensetzt, werden sich in dem Fall einer vollständigen Simulation noch geringere Schwankungen einstellen. Es muss allerdings betont werden, dass hier ausschließlich Abweichungen betrachtet werden, die ihren Ursprung in unterschiedlichen Initialisierungen der Sequenzgenerierung haben, dass also keine Aussagen über Abweichungen zwischen realen und generierten Daten gemacht werden.

Jahr	1	2	3	4	5	6	7	8	9
ERW	88721	55000	50421	44566	38701	33366	28157	22056	15915
STDA	1176	710	582	521	496	425	420	374	362
PROZ	1.33	1.29	1.15	1.17	1.28	1.28	1.49	1.70	2.27

Tabelle 6.3: *Erwartungswert und Schwankungen beim Generieren von Sequenzen am Beispiel summierter Spargeldeingänge basierend auf 500 Durchläufen mit je* 12 000 *Sequenzen*

6.3 Simulation eines realen Bausparkollektivs

Die in diesem und in dem vorhergehenden Kapitel gewonnen Erkenntnisse über die HMM-Modellierung von Bausparzeitreihen sollen nun zur Simulation eines realen Bausparkollektivs eingesetzt werden.

Trainingsdaten in der Kollektivsimulation
Verschiedene Kollektivsimulationen mit dem HMBM haben gezeigt, dass die Zusammensetzung der Trainingsmenge von großer Bedeutung für den Ausgang der Simulation ist. In der Mehrzahl der bisherigen Untersuchungen wurde der Abschlussjahrgang 1985 als Trainingsmenge eingesetzt, was zu guten Ergebnissen bei der Verlängerung älterer Abschlussjahrgänge geführt hat (s. Abschnitt 5.2.4). Versuche mit dieser Trainingsmenge, auch gemischte Bestände – also solche Datenmengen, die sowohl ältere als auch jüngere Jahrgänge enthalten – zu simulieren, haben gezeigt, dass die Beschränkung auf einen einzelnen Trainingsjahrgang wenig vorteilhaft ist. Der Grund hierfür ist darin zu sehen, dass die Häufigkeiten für bestimmte Spareraktionen sich im Zeitverlauf ändern können. Daher ist es für die Simulation eines gemischten Kollektives sinnvoller, auch eine gemischte Trainingsmenge einzusetzen.

Nach verschiedenen Versuchen hat sich eine zufällige Stichprobe, die sich aus Sequenzen mit einem Abschluss nach 1986 und aus Sequenzen mit erreichtem Tilgungsende bis einschließlich 1998 zusammensetzt, als geeignete Wahl für die Trainingsdaten herauskristallisiert. Der Vorteil dieser Trainingsmenge liegt darin, dass jüngere Entwicklungen im Verhaltensmuster der Sparphase abgebildet sind und dass gleichzeitig genügend Datenmaterial der Darlehensphase vorliegt. Der Umfang der Stichprobe beträgt ca. 6000 Sequenzen und stellt somit eine gute Größe bezüglich der erreichbaren Trainingsqualität und bezüglich dem dafür notwendigen Berechnungsaufwand dar.

Modelltopologie und Modelldimension
Zur Durchführung von Kollektivsimulationen wurden die Topologien „SLR0“ nach Abbildung 4.3 (S. 68) und „BSPK“ nach Abbildung 5.17 (S. 116) verwendet.

Die zur Abbildung realer Sequenzen sehr gut geeignete BSPK-Topologie (5.4.4) hat bei der Verlängerung bestehender und bei der Generierung künstlicher Sequenzen folgenden gravierenden Nachteil gegenüber dem Modell „SLR0“ gezeigt: Bei nicht zu vernachlässigend vielen Sequenzen wurde eine Zuteilung generiert, ohne dass die notwendigen Zuteilungsvorraussetzungen (s. Abschnitt 4.1.1) hierfür erfüllt wären. Dies wiederum hat negative Auswirkungen auf fast alle weiteren Kollektivzeitreihen. Dieser Umstand kann folgendermaßen erklärt werden: Befindet sich das Modell im Zustand „Z“ (Zuzahlung), so erfolgt im nächsten Zeitschritt zwingend ein Übergang in den Zuteilungszustand, auch dann, wenn die Zuteilungsbedingungen nicht erfüllt sind. Die Einführung der Übergangsklassen kann hieran nichts ändern, da der Übergang in die Zuteilung in allen Klassen mit Wahrscheinlichkeit eins geschieht. Da dieser unerwünschte Effekt beim Modell „SLR0“ nicht auftritt, wird im Folgenden ausschließlich diese Topologie betrachtet.

Weitere Rahmenbedingungen
Von den untersuchten Alternativen beim Training und bei der Simulation wird jeweils diejenige verwendet, die sich als am geeignetsten erwiesen hat. Die folgende Liste fasst dies noch einmal

zusammen und gibt damit die Rahmenbedingungen der nachfolgenden Simulation wieder.

Trainingsalgorithmus: MIX-HMM mit zufälliger Startpartition und zufälligen, aber hinreichend allgemeinen Initialmodellen; Wahl des Trainingsmodells aus 18 unterschiedlichen Trainingsläufen gemäß der besten Trainings-Likelihood pro Sequenz

Klassifikator: „MAW Distr“ (s. Abschnitt 6.1)

Bestandsbildung: Alle aktiven Sequenzen der Tarifklasse 3 des Jahres 1992; sämtliche Zeiträume nach 1992 werden ignoriert

Sequenzverlängerung und Sequenzgenerierung: Einsatz der Methode „EI_DISTR“ (s. Abschnitt 6.2)

Kollektivberechnung aus den Sequenzen
Zur Berechnung der Kollektivzeitreihen ist es notwendig, dass jede Sequenz mit einem Startzeitpunkt (Abschlussjahr) versehen ist. Aus den Sequenzen werden durch eine jahresweise Summation unter Berücksichtigung der individuellen Sequenzgewichtung (BS) die folgenden Kollektivzeitreihen berechnet:

- Anzahl aktiver Verträge
- gesamte BS
- zugeteilte BS
- nicht zugeteilte BS
- Spargeldeingang
- Tilgungen
- Beginn der Fortsetzung
- Beendigung der Fortsetzung
- Fortsetzerbestand
- Zuteilungen
- Darlehensnahmen
- Darlehensverzichte
- Kündigungen
- Abwicklungen durch vollständige Darlehenstilgung
- mittlere Anspargrade bei Zuteilung und Kündigung

Diese Größen werden aus den generierten Sequenzen und zum Vergleich aus den entsprechenden Realsequenzen berechnet und zur Bewertung der Simulation herangezogen.

Simulation des Gesamtkollektivs
Das simulierte Gesamtkollektiv setzt sich aus zwei Teilkollektiven zusammen:

1. Weiterentwicklung des zum Simulationsbeginn vorliegenden Bestandes
2. Nach Abschlussjahrgängen getrennt generierte neue Sequenzen (Neugeschäft)

Die Wahl des Simulationsbeginns zum Jahr 1993 stellt im gewissen Sinne einen Kompromiss zwischen dem Vorhandensein hinreichend langer Zeitreihen aus der Vergangenheit und einem genügend langen Vergleichszeitraum zur Bewertung der Simulationsergebnisse dar. Von der zugrunde liegenden Bausparkasse sind Daten der Jahre 1985 bis einschließlich 1998 vorhanden. Hieraus ergibt sich, dass zur Zuordnung der Anfangssequenzen bis zu acht Zeiträume genutzt werden können und dass ein sich über sechs Jahre erstreckender Vergleich von simulierten und realen Zeitreihen möglich ist.

Zur Generierung des Neugeschäftes wird jede Sequenz mit der gleichen Gewichtung von 30 TMD versehen, was in etwa der mittleren realen Bausparsumme entspricht. Es wird also keine Verteilung der Bausparsumme erzeugt. Das Gesamtvolumen der einzelnen Neugeschäftsjahrgänge wird aus den (bekannten) Realdaten ermittelt. Hierdurch sind ein Vergleich von Real- und Simulationszeitreihen und eine Bewertung der Simulation möglich, die nicht von eventuellen Abweichungen der Neugeschäftsvolumina verfälscht werden. Im Einsatz des Modells zur Simulation der zukünftigen Kollektiventwicklung muss das Neugeschäftsvolumen vorgegeben werden und ist somit ein freier Parameter der Simulation.

Die Abbildungen 6.2 bis 6.6 zeigen eine Auswahl der oben genannten Kollektivzeitreihen gewonnen aus den realen und den simulierten Sequenzen. Eine Bewertung dieser Resultate und weiterer Eigenschaften des HMBM wird im nachfolgenden Abschnitt 6.4 vorgenommen.

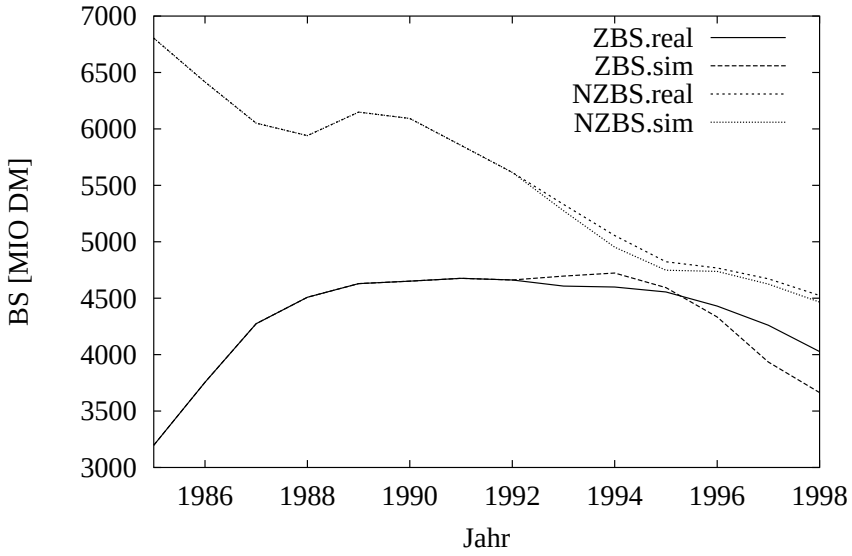

Abbildung 6.2: *Reale und simulierte Zeitreihen der zugeteilten Bausparsummen (ZBS) und der nicht zugeteilten Bausparsummen (NZBS); Simulationsbeginn: 1993*

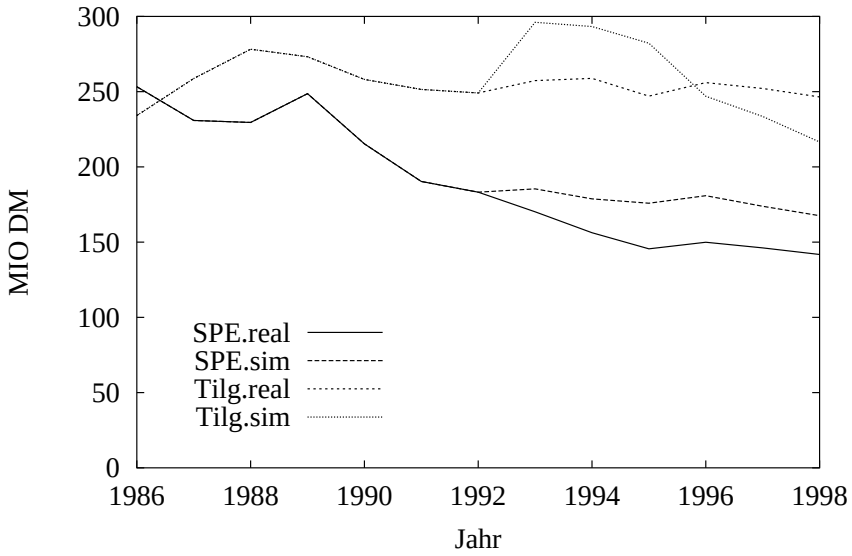

Abbildung 6.3: *Reale und simulierte Zeitreihen der Spargeldeingänge (SPE) und der Tilgungen (Tilg); Simulationsbeginn: 1993*

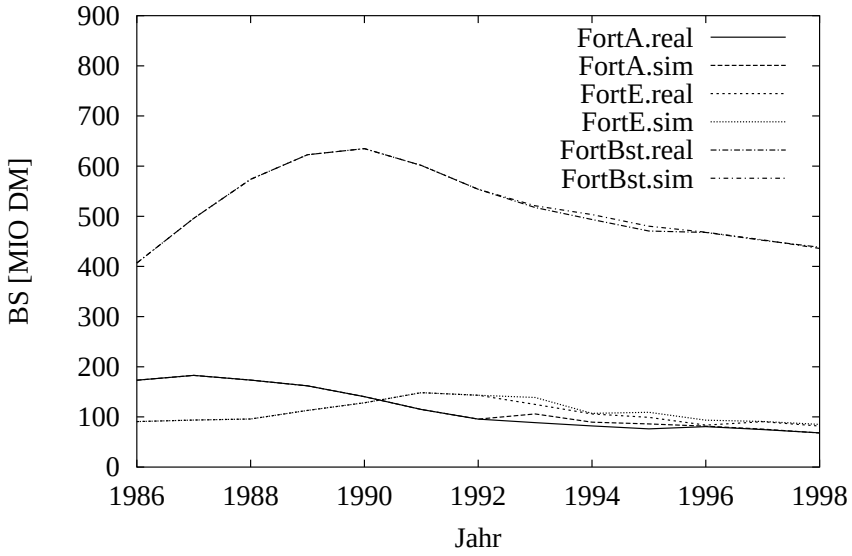

Abbildung 6.4: *Reale und simulierte Zeitreihen der Fortsetzungen: angefangene Fortsetzungen (FortA), beendete Fortsetzungen (FortE) und Bestand an Fortsetzungen (FortBst); Simulationsbeginn: 1993*

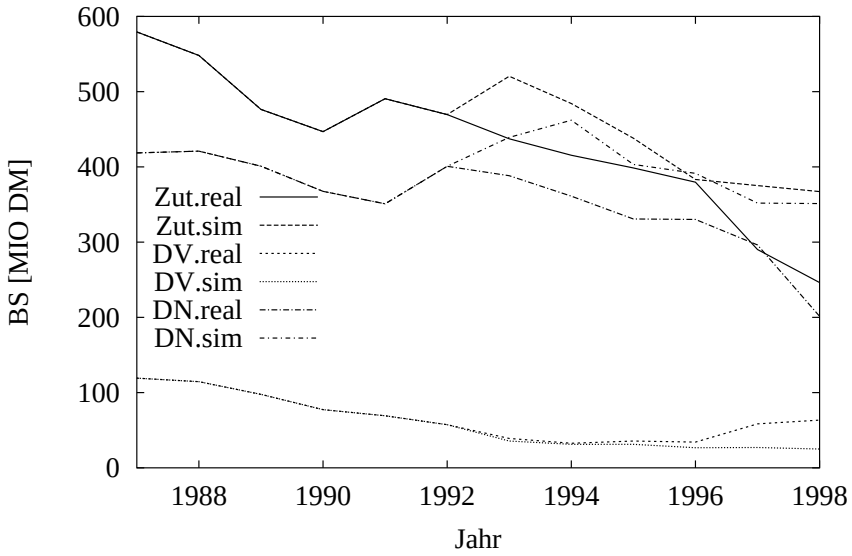

Abbildung 6.5: *Reale und simulierte Zeitreihen der Zuteilungen (Zut), der Darlehensverzichte (DV) und der Darlehensnahmen (DN); Simulationsbeginn: 1993*

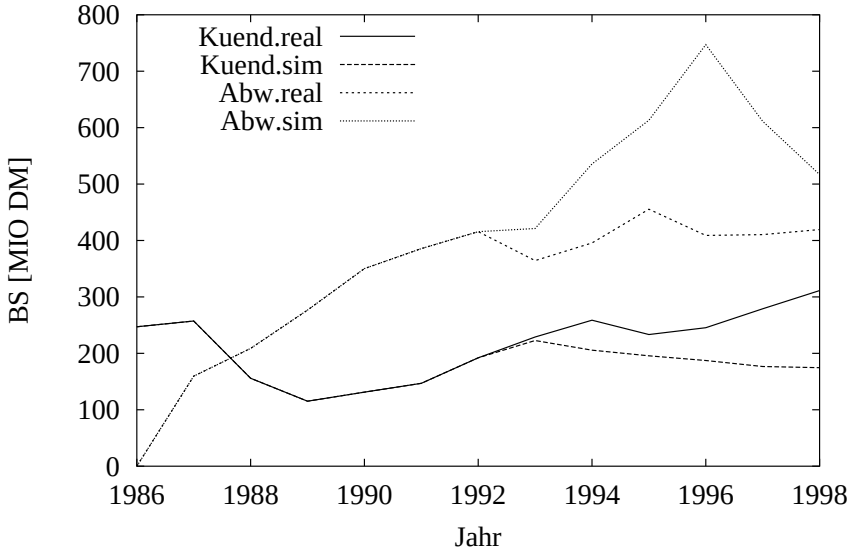

Abbildung 6.6: *Reale und simulierte Zeitreihen der Kündigungen (Kuend) und der Abwicklungen durch Tilgungsende (Abw); Simulationsbeginn: 1993*

6.4 Bewertung des HMBM

Kollektivsimulation

Nachfolgend sind zwei Listen aufgeführt, die die positiven und die negativen Eigenschaften des

entwickelten HMBM zusammenfassen.

Vorteile:

- Das Modell basiert auf Einzelvertragsdaten; daher ist prinzipiell eine sehr genaue Abbildung des realen Bausparkollektivs gegeben.
- Das Modell ist sehr gut in der Lage, unterschiedlich dimensionierte Daten (unterschiedlich lange Zeitreihen) zu verarbeiten. Die in geometrischen Clusterverfahren vorhandene Problematik der Definition eines sinnvollen Abstandes zwischen Vektoren unterschiedlicher Dimension entfällt. Statt dessen werden a posteriori Modellwahrscheinlichkeiten berechnet, die sich für beliebige Sequenzlängen angeben lassen.
- Die Abbildung der statistischen Eigenschaften der Daten, die Klassifikation der Sequenzen und die Generierung neuer Daten sind einer einheitlichen Beschreibung in Form des trainierten Modells unterworfen.
- Das Modell ist erweiterbar. Die Trainingsdaten können auf mehrdimensionale Zeitreihen und auf die Kombination von Zeitreihen mit zeitunabhängigen Merkmalen (z.B. demografische Daten) erweitert werden.
- Der Ablauf eines Bausparvertrages läßt sich sehr anschaulich in eine korrespondierende HMM-Topologie übertragen.
- Das Modell stellt ein gute Parametrisierung der statistischen Eigenschaften der Trainingsdaten dar. Dies zeigt sich beim Vergleich generierter Zeitreihen mit den entsprechenden Real-Zeitreihen. Beim Generieren eines einzelnen Abschlussjahrganges ergeben sich bessere Resultate als bei der Erzeugung eines Gesamtkollektivs. Das Gesamtkollektiv ist aufgrund zeitlicher Veränderungen der Sparer-Verhaltensmuster heterogener als ein einzelner Abschlussjahrgang.
- Es werden für das Training des Modells moderate Laufzeiten und für die Generierung künstlicher Sequenzen sehr kurze Laufzeiten benötigt (s. nachfolgenden Abschnitt).
- Als Nebeneffekt kann das Modell zur Analyse der Daten im Hinblick auf Datenfehler (z. B. unzulässige Abfolge der drei Phasen eines Bausparvertrages) eingesetzt werden.

Nachteile:

- Aufgrund lokaler Maxima gibt es keine absolute Garantie dafür, dass der Clusteralgorithmus ein „gutes“ Modell liefert. Es ist keine obere Schranke für die optimale Lösung bekannt und damit ist unklar, wie weit die gewonnene Lösung vom Optimum entfernt ist.
- Das Verfahren weist eine geringe Transpararenz auf; es ist nur schwer möglich, auf das Resultat steuernd Einfluss zu nehmen. Dies wäre bei der Bauspar-Modellierung wünschenswert, wenn für bestimmte zukünftige Verhaltensmuster veränderte Wahrscheinlichkeiten gegenüber der Vergangenheit zu erwarten sind.
- Bei relativ selten auftretenden Aktionen (z.B. Darlehensverzicht) ergeben sich z. T. recht deutliche Abweichungen zwischen generierten und realen Zeitreihen.
- Die verwendete Trainingsmenge ist mit Bedacht zu wählen, da das trainierte Modell direkt die statistischen Eigenschaften der Trainingsmenge widerspiegelt.

Hieraus wird deutlich, dass die Vorteile überwiegen und dass die Modellierung von Bausparkollektiven erfolgversprechend mit Hidden-Markov-Modellen durchgeführt werden kann.

Laufzeiten
Die nachfolgenden Angaben zu den Laufzeiten beziehen sich auf die in C programmierte HMM-Bibliothek und einen Compac Tro64 Server mit vier dec alpha EV6. 7 (667 MHz) Prozessoren und sechs GByte Hauptspeicher, außer bei der Sequenzerzeugung, bei der ein Sun E 450 Server mit vier Sun Ultra Sparc 2 (400 MHz) Prozessoren und einem GByte Hauptspeicher zum Einsatz kam.

Alle nachfolgenden Zeitangaben sind tatsächliche Laufzeiten des jeweiligen Programms („real time“); die Unterschiede zu den CPU-Zeiten („user time“) sind äußerst gering, da der Rechner zu den Zeitpunkten der Zeitmessung nicht ausgelastet war. Die betrachteten Instanzen besitzen eine für die Kollektivsimulation einer Tarifklasse typische Größendordnung.

Training: Die Laufzeit hängt stark von der Anzahl der benötigten Baum-Welch-Iterationen ab. In der Tabelle 6.4 sind gemittlte Werte über jeweils sechs Trainingsläufe bei einer Trainingsmenge von 6500 Sequenzen angegeben.

Modelltopologie	Algorithmus	Anzahl Modelle	Laufzeit (s)
SLR0, 12 Zust.	HMM-Cluster	6	120. 29
SLR0, 12 Zust.	MIX-HMM	6	409. 43
BSPK, 13 Zust.	MIX-HMM	5	372. 63

Tabelle 6.4: *Mittlere Trainingszeiten bei einer Trainingsmenge von 6500 Sequenzen*

Zu beachten ist, dass die angegebenen Zahlen Mittelwerte eines einzelnen Trainingslaufes sind. Da der Trainingsalgorithmus jedoch nur ein lokales Optimum findet, ist es sehr ratsam mehrere Trainingsdurchgänge mit unterschiedlicher Initialisierung vorzunehmen, wodurch sich die Laufzeit um den entsprechenden Faktor erhöht.

Laufzeiten für das ebenfalls untersuchte Simulated Annealing liegen im Bereich von mehr als 3000 Sekunden (s. Abbildung 5.12) für das Training eines einzelnen Modells mit 18 Zuständen. Es ist zu erwarten, dass die Laufzeiten des Simulated Annealing für ein aus mehreren Modellkomponenten bestehendem Modell um einige Größenordnungen über den erzielten Zeiten der Baum-Welch basierten Clusteralgorithmen liegen.

Beim HMM-Cluster-Verfahren muss jedes Modell im Mittel mit K/C Sequenzen trainiert werden, wenn K die Gesamtanzahl der Sequenzen und C die Anzahl der Modelle bezeichnen. Das MIX-HMM-Verfahren dagegen trainiert jedes Modell mit K Sequenzen. Die Laufzeit einer einzelnen Baum-Welch-Iteration ist linear in der Anzahl der Sequenzen. Vorausgesetzt, die Anzahl der Baum-Welch-Iterationen ist bei beiden Verfahren im Mittel gleich, dann folgt, dass die mittlere Laufzeit des MIX-HMM-Algorithmus um den Faktor C über der vom HMM-Cluster-Algorithmus liegt. Im obigen Beispiel wurde dagegen nur ein Faktor von 3. 4 (anstatt sechs) gemessen. Die Ursache hierfür ist eine Beschleunigung im MIX-HMM-Algorithmus, die dafür sorgt, dass Sequenzen mit sehr kleiner Komponentenwahrscheinlichkeit bezüglich eines bestimmten Modells, nicht zum Training dieses Modells herangezogen werden.

Sequenzerzeugung: Erzeugung aller Sequenzen der untersuchten Tarifklasse TK 3 (528 000 Sequenzen) aus dem Gesamtdatensatz:
168. 54 Sekunden.

Bestandsbildung: Klassifikation und Verlängerung von 229 000 Bestandssequenzen:
104. 36 Sekunden.

Neugeschäftsgenerierung: Neugeschäft von 1993 bis 1998 (86710 Sequenzen) mit einer maximalen Länge von $T = 15$: 13. 41 Sekunden.

Kapitel 7

Zusammenfassung und Ausblick

In dieser Arbeit wurde mit dem Hidden-Markov-Bauspar-Modell (HMBM) ein stochastisches Modell zur Simulation von Zeitreihen aus einem Bausparkollektiv entwickelt. Das Modell setzt sich aus mehreren Hidden-Markov-Modellen zusammen, die unter Verwendung realer Daten individueller Bausparverträge trainiert wurden, um somit die statistischen Eigenschaften eines Bausparkollektivs zu erfassen. Die trainierten Modelle konnten eingesetzt werden, um künstliche Sequenzen zu generieren, deren Überlagerung die simulierte zukünftige Kollektiventwicklung ergibt.

Zum Training der Modelle wurden zwei unterschiedliche modellbasierte Clusterverfahren entwickelt und untersucht. Ziel dieser Verfahren ist es, die Modellparameter zu ermitteln, die eine vorgegebene Zielfunktion maximieren. Der HMM-Cluster-Algorithmus ist ein iterativer Algorithmus, der das Modelltraining unter Verwendung geeigneter Sequenzpartitionen durchführt, während das MIX-HMM-Verfahren die Hidden-Markov-Modelle als Komponenten eines so genannten Mischmodells auffasst, dessen Parameter unter Verwendung eines allgemeinen Optimierungsschemas (EM-Algorithmus) festgelegt werden. Beim Training der Hidden-Markov-Modelle wurden Vorgaben gemacht, woraus unterschiedliche Trainingsvarianten entstanden, die eingehend untersucht und miteinander verglichen wurden. Da beide Clusterverfahren im Allgemeinen nicht in der Lage sind, ein globales Optimum zu finden, wurden verschiedene Verfahren zur Verbesserung des lokalen Optimums entwickelt und getestet.

Das HMBM ermöglicht eine einheitliche Beschreibung des gesamten Ablaufes eines Bausparvertrages. Neben den Spar- und Tilgungszahlungen konnten die wichtigsten Aktionsmöglichkeiten, die ein Bausparer während der Vertragslaufzeit besitzt, im Modell berücksichtigt werden. Hierzu wurden spezielle Zustände eingeführt und die Einhaltung der generellen Abfolge dieser Zustände konnte durch eine geeignete Modelltopologie sichergestellt werden.

Desweiteren wurde die in [23] entwickelte Erweiterung der Hidden-Markov-Modelle um so genannte Übergangsklassen verfeinert und auf den gesamten Vertragsablauf ausgedehnt. Damit war es möglich, wichtige bauspartechnische Nebenbedingungen beim Generieren künstlicher Sequenzen einzuhalten. Bei der Modellierung eines Bausparkollektivs spielen die Bausparsummen der zugrundeliegenden Verträge eine entscheidende Rolle. Sie können als Gewichte interpretiert werden, da sich aus ihnen die Bedeutung der jeweiligen Verträge für das Gesamtkollek-

tiv ergibt. Aus diesem Grund wurden die oben erwähnten Clusterverfahren so erweitert, dass die Clusterung gewichteter Objekte möglich wurde.

Mit dem Modellansatz wurde eine Kollektivsimulation eines Teils einer realen Bausparkasse durchgeführt. Der Simulationsbeginn wurde in die Vergangenheit zurückverlegt, um somit die Möglichkeit des Vergleichs zwischen realen und simulierten Kollektivgrößen zu haben. Insgesamt konnte eine gute Übereinstimmung zwischen Simulation und Realität erzielt werden, wenngleich bei bestimmten Größen noch signifikante Abweichungen festzustellen waren. Hier besteht bei einer Weiterentwicklung des Modells noch Spielraum für Verbesserungen. Als Fazit kann festgehalten werden, dass das in dieser Arbeit entwickelte HMBM einen vielversprechenden Modellansatz zur Analyse und Simulation von Finanzzeitreihen aus Bausparkollektiven darstellt.

Die durchgeführten Entwicklungen und Untersuchungen bieten Anknüpfungspunkte für verschiedene Weiterentwicklungen. So wäre es interessant, die Trainingssequenzen auf mehrdimensionale Zeitreihen zu erweitern und zu untersuchen, ob hieraus verbesserte Simulationsergebnisse resultieren. Eine Verknüpfung der Zeitreihe mit zeitunabhängigen Merkmalen (z. B. demographische Daten) könnte hergestellt werden. Ein weiterer Punkt, der in dieser Arbeit ansatzweise behandelt wurde, ist die Bestimmung einer geeigneten Modelltopologie aus den Daten heraus. Hierzu wäre es reizvoll, ein Lernverfahren zu entwickeln, dass auf einem Bayesschen Ansatz basiert.

Erste Versuche, das entwickelte Modell zur Simulation von Aktienkursen einzusetzen waren nicht sehr erfolgreich. Dies schließt jedoch nicht aus, dass weitere Anstrengungen in diese Richtung lohnenswert sein könnten und zu neuen Einsichten über die Mechanismen des Börsengeschehens führen könnten.

In der praktischen Anwendung des Modells, als Basis für planerische Aufgaben, wäre die Steuerung der Simulation und Berechnung unterschiedlicher Szenarien von Bedeutung. Wenn konkrete Erwartungen bezüglich zukünftiger Entwicklungen bestimmter Verhaltensmuster vorhanden sind, die durch die Trainingsdaten nicht zu decken sind, müßte eine Möglichkeit geschaffen werden, dies in das Modell zu integrieren. Dies geht eng einher mit der Aufgabe, externe Größen wie beispielsweise das generelle Zinsniveau mit dem Modell zu verknüpfen.

Literaturverzeichnis

[1] R. Azencott. *Simulated annealing*. Wiley, New York, 1992.

[2] L. E. Baum and T. Petrie. Statistical inference for probabilistic functions of finite Markov chains. *Ann. Math. Statist.*, 37:1554–1563, 1966.

[3] L. E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Ann. Math. Statist.*, 41:164–171, 1970.

[4] B. Bertsch, B. Hölzle, and H. Laux. *Handwörterbuch der Bauspartechnik*. Verlag Versicherungswirtschaft, Karlsruhe, 1998.

[5] J. A. Bilmes. A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. Technical Report TR-97-021, International Computer Science Institute, Berkeley, CA, 1998.

[6] I.N. Bronstein and K.A. Semendjajew. *Taschenbuch der Mathematik*. Harry Deutsch, Frankfurt (Main), 1987.

[7] J. Cavanaugh. A large-sample model selection criterion based on kullback's symmetric divergence. *Statistics and Probability Letters*, 44:333–344, 1999.

[8] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Statist. Soc. Ser. B*, 39(1):1–38, 1977.

[9] R.J. Elliot, L. Aggoun, and J.B. Moore. *Hidden Markov Models*. Springer, New York, 1995.

[10] G. S. Fishman. *Monte Carlo – Concepts, Algorithms and Applications*. Springer, New York, 1996.

[11] M. Fridman. A two state capital asset pricing model. Technical report, University of Minnesota, Minnesota, 1994.

[12] F. Gotterbarm. *Modelle und Optimierungsansätze zur Analyse des kollektiven Bausparens*. PhD thesis, Universität Bonn, Bonn, 1985.

[13] J.A. Hartigan and M.A. Wong. Algorithm AS 136: A k–means clustering algorithm. *Applied Statistics*, 28:100–108, 1978.

[14] H. Heuser. *Lehrbuch der Analysis*. Teubner, Stuttgart, 1998.

[15] X. D. Huang, Y. Ariki, and M. A. Jack. *Hidden Markov Models for Speech Recognition*. Edinburgh University Press, 1990.

[16] S. Ingrassia. A comparison between the simulated annealing and the EM algorithms in normal mixture decompositions. *Statistics and Computing*, 2:203–211, 1992.

[17] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, Englewood Cliffs, New Jersey, 1988.

[18] B.-H. Juang. Maximum-likelihood estimation for mixture multivariate stochastic observations of Markov chains. *AT&T Tech. J.*, 64(6, part 1):1235–1249, 1985.

[19] R.E. Kass and A.E. Raftery. Bayes factors. *Journal of the American Statistical Association*, 90:773–795, 1995.

[20] A. Kehagias. Bayesian classification of hidden Markov models. *Math. Comput. Modelling*, 23(5):25–43, 1996.

[21] I. Kellershohn. *Mathematische Simulation von Bausparkollektiven mit Hilfe von empirischen Verteilungen in monothetischen hierarchischen Kollektivclusterungen*. PhD thesis, Universität zu Köln, Köln, 1992.

[22] S. Kirkpatrick, C. D.Jr. Gelatt, and M.P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

[23] B. Knab. *Erweiterungen von Hidden–Markov–Modellen zur Analyse ökonomischer Zeitreihen*. PhD thesis, Universität zu Köln, Köln, 2000.

[24] B. Knab, R. Schrader, I. Weber, K. Weinbrecht, and B. Wichern. Mesoskopisches Simulationsmodell zur Kollektivfortschreibung. Technical Report ZPR97-295, Mathematisches Institut, Universität zu Köln, 1997.

[25] A. Krogh, M. Brown, I.S. Mian, K. Sjölander, and D. Haussler. Hidden Markov models in computational biology. *Journal of Molecular Biology*, 235:1501–1531, 1994.

[26] H. Laux. *Die Bausparfinanzierung*. Verlag Recht und Wirtschaft, Heidelberg, 1992.

[27] S. E. Levinson, L. R. Rabiner, and 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.

[28] L. A. Liporace. Maximum likelihood estimation for multivariate observations of Markov sources. *IEEE Trans. Inform. Theory*, 28(5):729–734, 1982.

[29] G.J. McLachlan and K.E. Basford. *Mixture Models: Inference and Applications to Clustering*. Marcel Dekker, Inc., New York, Basel, 1988.

[30] G.J. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. John Wiley & Sons, Inc, New York, 1997.

[31] A. Neath and J. Cavanaugh. Regression and time series model selection using variants of the Schwarz information criteria, 1997.

[32] L.R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–285, 1989.

[33] P.A. Schrodt. Pattern recognition of international crisis using hidden Markov models. Technical report, University of Kansas, Lawrence, KS, 1997.

[34] G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6:461–464, 1978.

[35] P. Smyth. Clustering using Monte Carlo Cross-Validation. *Proc. of the 2nd. Int. Conf. on Knowledge Discovery and Data Mining*, pages 126–133, 1996.

[36] P. Smyth. Model selection for probabilistic clustering using cross-validated likelihood. Technical Report TR-98-09, University of California, Irvine, 1998.

[37] P. Smyth. Probabilistic model-based clustering of multivariate and sequential data. Technical report, University of California, Irvine, 1999.

[38] P. Smyth. A general probabilistic framework for clustering individuals. Technical Report TR-00-09, University of California, Irvine, 2000.

[39] A. Stolcke and S. M. Omohundro. Best-first Model Merging for Hidden Markov Model Induction. Technical Report TR-94-003, International Computer Science Institute, Berkeley, CA, January 1994.

[40] H. Trebbe. The Münster Tagging Project – Errata in Rabiner's HMM-Tutorial. Arbeitsbereich Linguistik, Westfälische Wilhelms-Universität, Münster, 1995.

[41] I. Vannahme. *Clusteralgorithmen zur mathematischen Simulation von Bausparkollektiven*. PhD thesis, Universität zu Köln, Köln, 1996.

[42] C.F.J Wu. On the convergence proporties of the EM algorithm. *Annals of Statistics*, 11:95–103, 1983.

Danksagung

Viele Menschen haben mich auf unterschiedlichste Art und Weise beim Erstellen dieser Arbeit unterstützt. Ganz besonders bedanken möchte ich mich bei

- Prof. Dr. R. Schrader, der durch die Betreuung der Arbeit und die Leitung des ZAIK mir diese Promotion erst ermöglicht hat
- Dr. Alexander Schliep, der mir zahlreiche Anregungen gegeben hat und der als Ansprechpartner ständig eine wertvolle Hilfe war
- Dr. Bernhard Knab für die tolle Zusammenarbeit und das konstruktiv kritische Korrekturlesen
- Dr. Barthel Steckemetz für die geistige Geburtshilfe und die Motivation, „am Ball zu bleiben“
- den Kolleginnen und Kollegen im Bauspar-Projekt für die interessante Zusammenarbeit im Projekt und den zugestandenen Freiraum besonders in der Endphase der Arbeit
- Nataliya Gayeva und Thordis Linda Thorarinsdotir für die tatkräftige Unterstützung
- allen weiteren aktuellen und vielen ehemaligen Kolleginnen und Kollegen am ZAIK, die dazu beigetragen haben, dass ich gerne an die Zeit am Institut zurück denken werde
- meiner Familie, die mir das Studium und die Verwirlichung meiner Interessen ermöglicht hat
- Dr. Annette Vielhauer für die große Geduld, die moralische Unterstützung und das Korrekturlesen.

Kurzzusammenfassung

Hidden-Markov-Modelle (HMM) finden in unterschiedlichen Anwendungsgebieten der stochastischen Modellierung ihre Anwendung. Besonders in der automatischen Spracherkennung und in der Bioinformatik wird dieser Modelltyp erfolgreich eingesetzt. In dieser Arbeit wird der HMM-Ansatz zur Beschreibung von Finanzzeitreihen eines Bausparkollektivs untersucht. Die betrachteten eindimensionalen Zeitreihen bilden Aktionen einzelner Bausparer ab. Mit dem Hidden-Markov-Modell ist es möglich, relevante Strukturen und statistische Kenngrößen der Zeitreihen zu parametrisieren (Modelltraining) um somit Aussagen über die zeitliche Weiterentwicklung des Bausparkollektivs zu treffen. Hierzu werden zwei hmm-basierte Clusterverfahren entwickelt und verglichen. Insbesondere die Kombination von HMMen mit einem Mischmodell stellt einen vielversprechenden Ansatz in der Simulation von Bausparkollektiven dar. Mit diesem Modellansatz ist es möglich, Zeitreihen unterschiedlicher Dimension miteinander zuvergleichen und zu verarbeiten ohne ein explizites Abstandsmaß zu definieren. Dies ist ein bedeutender Vorteil gegenüber anderen Verfahren wie geometrischen Clusteralgorithmen. Es werden eingehend verschiedene Aspekte des Modelltrainings untersucht und bewertet. Mit Hilfe eines Pseudo-Zufallszahlengenerators können die trainierten Modelle direkt zur Erzeugung künstlicher Zeitreihen eingesetzt werden. Ein Ergebnis dieser Arbeit ist, dass es mit entsprechend trainierten Hidden-Markov-Modellen möglich ist, relevante Zeitreihen eines Bausparkollektivs zu generieren, die in sehr guter Übereinstimmung mit den entsprechenden realen Zeitreihen sind. Hierzu werden einige Anpassungen und Erweiterungen des Basis-HMMs hergeleitet.

Abstract

Hidden-Markov-Models (HMM) have been widely applied in different areas of stochastic modeling. Especially in the field of automatic speech recognition and bioinformatics this kind of modeling has been proved to be very successful. Here the description and simulation of economic time series arising from a loan bank using HMMs is investigated. The considered one dimensional time series represent transactions of individual customers. With the use of Hidden-Markov-Models it is possible to parameterize relevant structures and the statistics of the data (training of the model) to make predictions concerning the future development of the loan bank. For this purpose two HMM-based cluster methods are developed and compared. Particularly the combination of HMMs with a mixture model is a promising approach in modelling loan banking time series. With this kind of model it is possible to compare and process time series data of different dimension without defining an explicit distance measure. This is a significant advantage over other methods like geometric clustering algorithms. Different aspects affecting the training of the models are compared and evaluated. With the aid of a pseudo random number generator it is straight forward to use the trained models as generators for artificial time series. It is demonstrated that it is possible to produce sequences in very good agreement with the corresponding real data with suitably trained HMMs. To achieve this, some adoptions and extensions of the basic HMM are derived.

Lebenslauf

Persönliche Daten

Name:	Bernd Wichern
Adresse:	Marsiliusstraße 69, 50937 Köln
Geburtsdatum:	11. Dezember 1966
Geburtsort:	Zeven
Familienstand:	ledig
Staatsangehörigkeit:	deutsch

Ausbildung/Studium

1973–1977	Grundschule, Sittensen
1977–1979	Orientierungsstufe, Sittensen
1979–1986	Sankt Viti Gymnasium Zeven, Abschluss Abitur
1986–1988	Zivildienst, Ev. Hospital Lilienthal
1988–1995	Studium der Physik, Universität Oldenburg
6/1995	Diplom in Physik

Berufstätigkeit

9/1995 – 11/1995	Wissenschaftliche Hilfskraft an der Univ. Oldenburg
12/1995–6/2001	Wissenschaftlicher Mitarbeiter am Mathematischen Institut / Zentrum für Angewandte Informatik Köln, Universität zu Köln

Köln, November 2001

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.