Lehrstuhl für Mensch-Maschine-Kommunikation
Technische Universität München

Erzeugung robuster akustisch-phonetischer Modelle für die automatische Spracherkennung durch explizite Gruppenbildungen

Robert Faltlhauser

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs

genehmigten Dissertation.

Vorsitzender		Univ.-Prof. Dr.-Ing. J. Eberspächer
Prüfer der Dissertation:	1.	Apl. Prof. Dr.-Ing., Dr.-Ing. habil. G. Ruske
	2.	Univ.-Prof. Dr.-Ing. habil. R. Hoffmann Technische Universität Dresden

Die Dissertation wurde am 8.5.2002 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 14.11.2002 angenommen.

Zusammenfassung

Bei Systemen zur automatischen Spracherkennung stellt die Musterrepräsentation mittels der stochastischen Hidden-Markov-Modelle den aktuellen Stand der Technik dar. Die Parameter dieser Modelle werden anhand großer Sprachkorpora geschätzt. Im Mittel sind derartig trainierte Modelle sehr leistungsfähig - für eine konkrete Spracheingabesituation sind sie jedoch u.U. nicht optimal angepasst. Tendenziell enthalten Trainingskorpora jedoch Sprachmuster von sehr vielen Sprechern in unterschiedlichsten Sprachsituationen. Hierunter lassen sich Beispiele finden, die einer konkreten Spracheingabe ähnlich sind.

In dieser Arbeit wurden Methoden untersucht, wie sich bereits aus den Trainingsdaten charakteristische, spezialisierte HMM-Modelle ableiten lassen. Als Kernkonzept wurde hierbei das Prinzip der Gruppenbildung zugrundegelegt und untersucht. Die explizite Ausbildung von Gruppen ermöglicht einen Mittelweg zwischen der sicheren Schätzung der Modellparameter einerseits, und einer spezifischeren Modellierung andererseits. Als mögliche Ziele der Anpassung wurden schwerpunktmäßig die Variation der Sprecher sowie deren veränderliche Sprechgeschwindigkeit näher betrachtet.

Das Grundprinzip der Gruppenbildung findet sich bereits bei unspezialisierten Modellen auf verschiedenen Ebenen der stochastischen Modellierung. So werden auf unterster Ebene Gruppen von Mustervektoren gesucht, die sich durch eine gemeinsame Prototypverteilung repräsentieren lassen. Mit steigender Hierarchieebene nimmt die "Größe" der Einheiten zu. Die Prototypverteilungen sind Bestandteil der Codebücher innerhalb der HMM-Zustände. Deren Gruppierung zeigt sich insbesondere im Rahmen der kontextabhängigen Lautmodellierung als sinnvoll. Auf einer noch höheren Hierarchiestufe lassen sich komplette Modellsätze (oder Teile davon) zu Gruppen zusammenfassen.

Bei der Analyse des Sprechgeschwindigkeitseinflusses auf die stochastische Modellierung mit Gauss'schen Normalverteilungen zeigt sich eine direkte Abhängigkeit, die primär durch die Art der Merkmalsvektorbildung induziert wird. Insbesondere die Berechnung der Delta-Koeffizienten führt zu einer hohen Korrelation zwischen mittlerem Score und Sprechgeschwindigkeit bzw. mittlerem Score und Phonemdauer. Diese Abhängigkeit kann im Gegenzug dazu genutzt werden, um in der Anwendungsphase eine Aussage über die aktuelle Sprechgeschwindigkeit zu ermöglichen. Diese Information wird zur Selektion einer geeigneten Modellgruppe benötigt.

Bedingt durch die Deltaberechnung wirkt der Sprechgeschwindigkeitseinfluss insbesondere im transienten Bereich zwischen Phonemen. Die Modellierung der Kontextabhängigkeit von Phonemlauten zielt direkt auf die Erfassung dieser kontextspezifischen Variation der Musterverläufe. Daher wurde in diesem Zusammenhang insbesondere das phonetische Entscheidungsbaumverfahren zur Zustandsgruppierung auf seine Möglichkeiten zur sprechgeschwindigkeitsspezifischen Gruppenbildung hin untersucht. Hier zeigt sich, dass speziell die Integration der Sprechgeschwindigkeitsinformation in den Entscheidungsprozess in Form eines generalisierten Kontextes von Vorteil ist.

Zur Auffinden charakteristischer Sprechergruppen bietet sich der Einsatz automatischer Clusterverfahren, basierend auf expliziten Sprecherabstandsmaßen, an. Zur Definition der

Sprecherabstände wurden schwerpunktmäßig Gauss'sche Mixturmodelle auf ihre Verwendbarkeit hin untersucht. Die Abstände zwischen Sprechern können durch Likelihood-Betrachtungen definiert werden. Darüberhinaus kann die automatische Sprechergruppierung sehr robust in dem durch die Eigenvoices aufgespannten, reduzierten Subraum durchgeführt werden. Dem Eigenvoice-Ansatz liegt die Analyse der Inter-Sprecher Varianz der Modellparameter zugrunde. Durch Anwendung einer PCA-Transformation auf die zugehörige Inter-Sprecher Kovarianzmatrix der Modellparameter kann ein reduzierter Subraum erzeugt werden, in dem ein (bekannter) Sprecher durch sehr wenige Koeffizienten repräsentiert ist. Diese Koeffizienten erlauben eine Euklidische Abstandsdefinition, anhand derer eine robuste Gruppierung möglich wird.

Vorwort

Die vorliegende Arbeit entstand während meiner Zeit als wissenschaftlicher Mitarbeiter am Lehrstuhl für Mensch-Maschine-Kommunikation der Technischen Universität München. An dieser Stelle möchte ich mich bei allen bedanken, die das Entstehen dieser Arbeit ermöglicht haben. Hierbei gilt mein Dank in erster Linie meinem Doktorvater Herrn Prof. Dr.-Ing. Günther Ruske, ohne dessen Betreuung diese Arbeit nicht hätte verwirklicht werden können. Die von ihm geförderte freie Arbeitsweise, verbunden mit der stetigen Bereitschaft für fachliche und fachübergreifende Diskussionen, schuf die Basis für eine produktive wissenschaftliche Tätigkeit.

Herrn Prof. Dr.-Ing. Rüdiger Hoffmann vom Institut für Akustik und Sprachkommunikation der Technischen Universität Dresden möchte ich für die Übernahme des Zweitgutachtens meinen herzlichen Dank aussprechen.

Mein besonderer Dank gilt Herrn Prof. rer. nat. Manfred Lang für das Bereitstellen der Infrastruktur, ohne die die Untersuchungen in dieser Arbeit nicht hätten durchgeführt werden können.

Bedanken möchte ich mich in diesem Zusammenhang auch bei Herrn Prof. Hans G. Tillmann vom Institut für Phonetik und Sprachliche Kommunikation der Universität München für die Förderung im Rahmen des Graduiertenkollegs “Sprache, Mimik und Gestik” und den mir dadurch möglich gewordenen, fachübergreifenden Einblick in andere wissenschaftliche Disziplinen.

Meinen Dank sagen möchte ich all meinen Kollegen während dieser Zeit - insbesondere Herrn (mittlerweile Dr.) Thilo Pfau - für die Unterstützung und für wertvolle und fruchtbare Diskussionen. Den Herren Peter Brand und Heiner Hundhammer gebührt mein Dank für die Verwaltung des Rechnernetzes und die Möglichmachung meiner oftmaligen systemtechnischen Sonderwünsche.

Neben meinen Eltern gilt nicht zuletzt mein ganz besonderer Dank Herrn Claus-Peter Deglmann zusammen mit Frau Marianne Kuth für die fortwährende geistige und leibliche Unterstützung während dieser Zeit.

Miesbach, im April 2002

Robert Faltlhauser

Inhaltsverzeichnis

Kapitel 1

Einführung

1.1 Einführung und Motivation

Systeme zur automatischen Erkennung gesprochener Sprache haben, beim heutigen Stand der Technik, eine Schwelle erreicht, die sie für Praxisanwendungen tauglich erscheinen lässt. Im Alltagsleben beginnen solche Systeme bereits Einzug zu halten. Das Spektrum der Anwendungen ist breit gefächert und reicht von Diktiersystemen, über sprachgesteuerte Consumerprodukte (v.a. im Kfz), bis hin zu automatischen Callcentern. Aufgrund der gestiegenen Leistungsfähigkeit der Rechneranlagen, auf denen Spracherkennungssysteme zum Einsatz kommen, sind heutige Erkenner theoretisch in der Lage Vokabulargrößen von 100000 Wörtern echtzeitfähig und mit akzeptablen Fehlerraten zu verarbeiten.

Im praktischen Einsatz steigt die Fehlerrate von automatischen Spracherkennungssystemen jedoch stark an, wenn die Anwendungsbedingungen von den im Training des Systems gesehenen, ungestörten (Labor)Bedingungen abweichen. Die Palette der 'Störeinwirkungen' reicht hier von Hintergrundgeräuschen (z.B. Lüfter, Türschlagen, andere Sprecher), Charakteristika des Übertragungskanals (Telefon, Handy, Mikrofontyp) über dialektale Färbungen der Sprecher bis hin zu den Sprechgewohnheiten derselben. Gerade was Störeinflüsse anbelangt, weist der Mensch, bzw. sein Gehör und seine Wahrnehmung einige Fähigkeiten auf, die noch von keiner Maschine annähernd erreicht werden. Eine der hervorstechendsten ist hierbei wohl die Fähigkeit einem einzelnen Gesprächspartner auch bei hohem Hintergrundgeräuschpegel, z.B. in einer Gruppe von gleichzeitig sprechenden Menschen, zu folgen.

Aber selbst wenn sich die Umgebungsbedingungen nahe der Norm bewegen, haben dennoch insbesondere 2 unvermeidliche Aspekte maßgeblichen Einfluss auf die Erkennungsleistung von ASR-Systemen (ASR, engl.: 'Automatic Speech Recognition'), die in dieser Arbeit näher betrachtet werden sollen. Beide Gesichtspunkte resultieren aus den Stimmcharakteristika bzw. Sprechgewohnheiten des anwendenden Sprechers. Viele ASR-Systeme werden als sprecherunabhängige Erkenner konzipiert und sollten - theoretisch - für alle Sprecher gleich gut funktionieren. Nichtsdestoweniger unterscheiden sich verschiedene Sprecher mehr oder weniger stark in der Physiognomie ihres Sprechapparats, was zwangsläufig Unterschiede in der spektralen Zusammensetzung ihrer Stimme zur Folge hat. Dementsprechend arbeiten Erkennungssysteme - ohne etwaige Anpassungsmaßnahmen - per se für verschiedene Sprecher

unterschiedlich gut.

Darüber hinaus neigen Sprecher dazu ihre Aussprache und ihren Sprechstil während einer Äußerung zu variieren. Für dialektal artikulierende Sprecher ist es noch weitgehend möglich, diese Neigung bewusst zu unterdrücken und mit Normsprache ("Hochdeutsch") zu sprechen, um *sich* dem System anzupassen. Bei eher unterbewussten Angewohnheiten, wie beispielsweise dem Sprechstil, ist dies schwieriger und kann auf Dauer zum Nicht-Akzeptieren eines solches Systems führen. Der Sprechstil beinhaltet eine ganze Reihe von Parametern, durch die die Aussprache variiert werden kann. Zu den hervorstechendsten gehören beispielweise das Intonationsverhalten, das Setzen von Pausen und damit verbunden insbesondere die Sprechgeschwindigkeit. Gerade letztere hat maßgebliche Auswirkungen auf die Leistung von Spracherkennungssystemen. Die Veränderung der Sprechgeschwindigkeit äußert sich im Prinzip auf 2 Arten. Die "mittlere" Sprechgeschwindigkeit einerseits ist primär sprecherspezifisch geprägt und unterscheidet sich oftmals merklich zwischen Sprechern. Andererseits weist die Sprechgeschwindigkeit bei jedem Sprecher stark dynamische Züge auf, die meist durch die aktuelle Gesprächs- oder Gedankensituation verursacht werden. So sinkt gerade in Phasen, die mit ausgeprägten Überlegungen verbunden sind, die Sprechrate häufig deutlich ab.

Die in dieser Arbeit vorgestellten Verfahren widmen sich schwerpunktmäßig der Kompensation negativer Auswirkungen auf die Erkennungsleistung, die durch Sprecher und Sprechgeschwindigkeit verursacht werden.

Die Parameter von Spracherkennungssystemen werden anhand umfangreicher Sprachkorpora geschätzt. Große Trainingskorpora enthalten jedoch eine beträchtliche Variation - sowohl bezüglich der enthaltenen Sprecher, als auch hinsichtlich deren Sprechstils. Es lassen sich meist Beispiele in der Datenbasis finden, die der aktuellen Spracheingabesituation (Sprecher und Sprechgeschwindigkeit) ähnlich sind. Für die Erzeugung passender und robuster Modelle müssen allerdings ausreichend Beispieldaten vorliegen. Da dies jedoch häufig nicht der Fall ist, muss ein Mittelweg zwischen Genauigkeit und Menge der Beispieldaten beschritten werden. Als zentrale Maßnahme wird daher die gezielte Bildung charakteristischer (Modell)Gruppen untersucht. Ziel ist durch explizite Zusammenfassung ähnlicher Einheiten Informationen über Sprecher-, Sprechereigenschaften, sowie die Sprechgeschwindigkeit in die Gruppenbildung einfließen zu lassen, um die Spracherkennungsmodelle besser für den jeweiligen Anwender und seine Gewohnheiten vorzubereiten und damit letztendlich eine fehlerreduzierte Spracherkennung zu ermöglichen.

Als erster zentraler Punkt stellt sich in diesem Zusammenhang die Analyse und Erfassung der Auswirkungen auf die stochastische Modellbildung, wie sie durch die variable Sprechgeschwindigkeit verursacht wird. Hierbei werden insbesondere die Auswirkungen auf die sogenannte "akustische Modellierung" (vgl. Abb. 1.4) näher betrachtet. In einem zweiten Schritt wird die vorliegende Information durch gezielte Bildung von spezifischen Modellgruppen in die Spracherkennungsmodelle eingearbeitet. Die Gruppenbildung wurde auf verschiedenen Ebenen der akustischen Modellierung eines ASR-Systems untersucht, beginnend mit der probabilistischen Repräsentation mittels Wahrscheinlichkeitsdichtefunktionen, über die Auswahl und

Kombination der zu modellierenden Spracheinheiten, bis zur sprecher(gruppen)spezifischen Modellbildung.

In diesem Zusammenhang wurde mit dem Eigenvoice-Ansatz ein neuartiger Trainingsalgorithmus untersucht und implementiert, der es erlaubt, die Apriori-Information über die sprecherspezifischen Änderungen der Modellparameter als zusätzliche Wissensquelle bei der Generierung neuer Modelle hinzuziehen.

Das Konzept der Gruppenbildung ist ein grundlegender Mechanismus, der sich auf nahezu allen Ebenen der stochastischen Modellierung wiederfindet. Ausgangspunkt ist eine Menge von Einzelelementen, die aufgrund gemeinsamer Charakteristika, Merkmale oder Beziehungen zueinander in Gruppen eingeteilt bzw. zu Gruppen zusammengefasst werden. Für jede der eingeteilten Gruppen wird i.d.R. eine Modellrepräsentation (oder mehrere) erzeugt, die als Vergleichsreferenz in der Spracherkennungsphase dienen kann. Die Einteilung der Gruppen kann sowohl manuell als auch automatisch erfolgen. Die automatische Einteilung bietet sich an, wenn keine expliziten oder quantifizierbaren Attribute vorliegen. Bei dieser Betrachtungsweise hängt der Begriff der "Gruppe" eng mit dem Begriff der "Klasse" zusammen. Dieser beschreibt die Zieleinheiten einer Erkennung (=Klassifikation). Die Festlegung von definierten Klassen erfolgt i.d.R. manuell, unter Kenntnis der vorliegenden, definierten Attribute der Einzelobjekte. Die Klassenbildung kann dahingehend als "Teil" des Gruppenkonzepts aufgefasst werden.

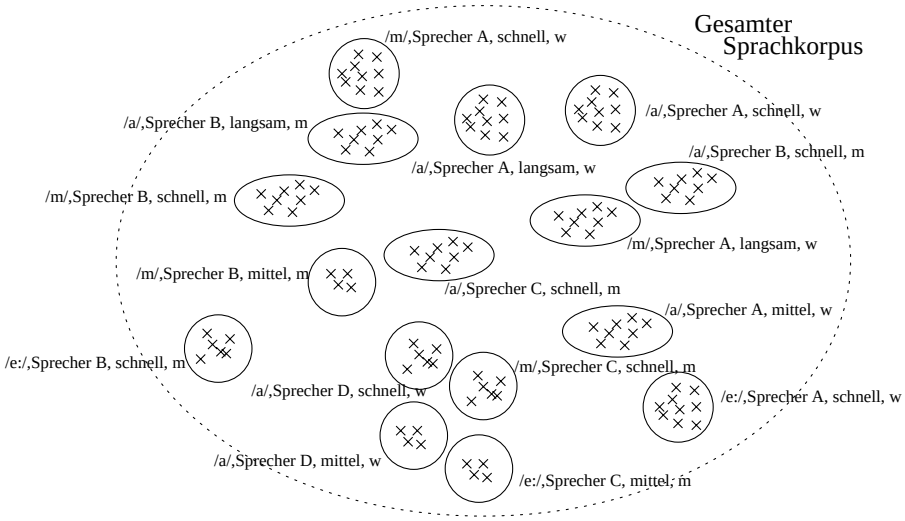

Abb. 1.1: Den einzelnen Merkmalsvektoren des Sprachkorpus lassen sich definierte Attribute zuordnen, anhand derer eine Einteilung erfolgen kann. Die Lage der Merkmalsvektoren erlaubt die Gewinnung von weiteren Informationen.

Die Vorverarbeitung (s. Abschnitt 1.4) eines Spracherkennungssystems erzeugt für die Sprachäußerungen des Trainingskorpus Sequenzen von Mustervektoren. Unterteilt man diese Vektorsequenzen in die artikulierten Worte und Wortuntereinheiten, so kann für jeden Mustervektor im Prinzip eine diesbezügliche Etikettierung erfolgen (s. Abb. 1.1). Die Information

über die jeweilige Wortuntereinheit stellt allerdings nur ein Attribut aus einer ganzen Reihe von Merkmalen dar. Weitere Attribute für einen Vektor können beispielsweise der zugehörige Sprecher sein, dessen Geschlecht, die Geschwindigkeit, oder die Emotion mit der der Vektor "geäußert" wurde. Diese weitergehenden Informationen müssen entweder gemessen, oder bereits bei der Aufnahme des Trainingskorpus mit registriert werden.

Über die fix quantisierten Charakteristika hinaus, lassen die einzelnen Mustervektoren auch Aussagen über Ähnlichkeitsbeziehungen zueinander zu. Diese können beispielsweise anhand geometrischer Abstandsmaße ermittelt werden. Von einer höheren Ebene aus betrachtet, kann mittels dieser "low-level" Abstände beispielsweise die Ähnlichkeit zwischen Sprechern bewertet werden. Noch nicht berücksichtigt ist, in Abb. 1.1, die eigentliche stochastische Modellierung mit Hidden-Markov-Modellen (HMM) (s. Abschnitt 1.2.2). Konventionellerweise wird bei der akustisch-phonetischen Modellbildung eine Repräsentation für eine Lauteinheit in Form eines parametrischen, stochastischen HMM-Modells erzeugt. Die Festlegung der zu klassifizierenden Lauteinheiten stellt hierbei bereits eine erste Klasseneinteilung dar. Die Unterteilung eines HMMs in Einzelzustände, sowie die Parameter dieses Modells lassen darüber hinaus eine Reihe weiterer Unterteilungen und Zuordnungen zu.

In Abb. 1.1 sind den dargestellten Vektoren eine Reihe von Attributen zugeordnet, wobei in dieser Arbeit eine Fokussierung auf die Attribute "Sprecher" und "Sprechgeschwindigkeit" erfolgt. Untersucht wurde, speziell im Fall der Sprechgeschwindigkeit, wie sich diese Information gewinnen, und als Zusatzinformation in die Gruppenbildungsprozesse integrieren lässt. Die Gruppenbildungen wurden auf den verschiedenen Ebenen der akustisch-phonetischen Modellierung betrachtet.

```
Menge der Lautmodelle (HMMs)
          einzelne HMMs bzw. HMM-Zustände
                    Codebücher mit Gauss'schen Prototypen
                              Mustervektoren
```

Die Gruppierung der Mustervektoren auf unterster Ebene wurde mittels sogenannter "Top-Down"-Clusterverfahren realisiert. Das Ziel hierbei ist, Vektorgruppen - "Ballungen" - im Merkmalsraum zu finden, die sich durch eine Gauss'sche Verteilung modellieren lassen. Da die Mustervektoren die Grundlage der akustischen Modellierung darstellen, wurde in Kapitel 3 schwerpunktmäßig untersucht, wie sich die Sprechgeschwindigkeit in den eigentlichen Mustervektoren niederschlägt. Hierbei zeigte sich, dass, bedingt durch die bei Spracherkennungssystemen typische Vorverarbeitung, die Merkmalsvektoren und die darauf aufbauende Modellierung hochgradig von der Sprechgeschwindigkeit beeinflusst sind. Durch geschickte Klasseneinteilung und Modellierung konnte ein Systemaufbau entwickelt werden, der es ermöglicht, die vorliegende Sprechrate aus den Merkmalen abzuschätzen. Aufbauend auf der gewonnenen Sprechgeschwindigkeitsinformation werden im nachfolgenden Kapitel Ansätze diskutiert, die diese Information bei der Gruppierung von HMM-Zuständen einbeziehen. Die Bestimmung charakteristischer Sprechergruppen ist zentrales Thema des letzten Kapitels. Hier wird eine auf dem Eigenvoice-Ansatz basierende Methode vorgestellt, die eine robuste Gruppierung und Zusammenfassung von Sprechern ermöglicht.

1.2 Automatische Spracherkennung

1.2.1 Dekodierungsgleichung

Der Ansatz zur automatischen Erkennung gesprochener Sprache, wie er beim heutigen Stand der Technik angewandt wird, ist vom Prinzip her gesehen ein Mustervergleich. Der direkte Vergleich von Test- und Referenzmustern mittels des DP-Algorithmus [Rus94] wird bei vielen Sprechern, aus Speicher- und Rechenzeitgründen unpraktikabel, da Referenzmuster für alle möglichen Artikulationsweisen von Wörtern gespeichert und verglichen werden müssten. Speziell für sprecherunabhängige Systeme hat es sich daher durchgesetzt, beim Mustervergleich die Referenzmuster durch eine stochastische Repräsentation zu ersetzen [Rab86, Rab89]. Die Parameter der stochastischen Modelle werden aus den Referenz(=Trainings)mustern geschätzt. Bei der Verwendung dieser Modelle stellt sich bei der Ermittlung der optimalen, hypothetischen Wortfolge das folgende Dekodierungsproblem:

$$\hat{\boldsymbol{w}} = \arg\max_{\boldsymbol{w}} p(\boldsymbol{w}|\boldsymbol{\mathcal{X}}) \tag{1.1}$$

Gesucht ist unter allen möglichen Wortfolgen $\boldsymbol{w} = \{w_1, w_2, \ldots, w_M\}$ der beliebigen Länge M diejenige Wortfolge $\hat{\boldsymbol{w}}$, die bezüglich der gegebenen Mustervektorfolge $\boldsymbol{\mathcal{X}} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_T\}$ die maximale Rückschlusswahrscheinlichkeit liefert. Unter Ausnutzung des Satzes von Bayes kann Gl. 1.1 umgeformt werden zu:

$$\hat{\boldsymbol{w}} = \arg\max_{\boldsymbol{w}} \frac{p(\boldsymbol{\mathcal{X}}|\boldsymbol{w})p(\boldsymbol{w})}{p(\boldsymbol{\mathcal{X}})} = \arg\max_{\boldsymbol{w}} p(\boldsymbol{\mathcal{X}}|\boldsymbol{w})p(\boldsymbol{w}) \tag{1.2}$$

Da der Term $p(\boldsymbol{\mathcal{X}})$ unabhängig von der gesuchten Wortfolge und damit für alle Wortfolgen gleich ist, kann er in Gl. 1.2 bezüglich der Maximierung entfallen. Es verbleiben die beiden Teilterme $p(\boldsymbol{\mathcal{X}}|\boldsymbol{w})$ und $p(\boldsymbol{w})$. $p(\boldsymbol{w})$ ist die sogenannte Sprachmodellwahrscheinlichkeit der Wortfolge $\boldsymbol{w}$. Zur Modellierung dieser Wahrscheinlichkeit wird ein Markov-Prozess N-ter Ordnung angenommen.

$$p(\boldsymbol{w}) = p(w_1) \prod_{i=2}^{M} p(w_i|w_{i-1}, w_{i-2}, \ldots, w_{i-N}) \tag{1.3}$$

Aus Gründen der Parameterschätzbarkeit beschränkt man sich im Rahmen des Sprachmodells meist auf Bi- $p(w_i|w_{i-1})$ oder Trigrammwahrscheinlichkeiten $p(w_i|w_{i-1}, w_{i-2})$. Die Wahrscheinlichkeiten selbst können anhand großer Sprachkorpora abgeschätzt werden. Allerdings sind effiziente Methoden zur Schätzung bzw. Interpolation [Kla98] nötig, da bereits bei einem Wortschatz von "nur" 10000 Wörtern $10000^2 = 10^8$ Bigrammwahrscheinlichkeiten (bzw. 10^{12} Trigramme) bestimmt werden müssten. Selbst bei sehr großen Sprach- bzw. Textkorpora mit Millionen von Wörtern kann nicht davon ausgegangen werden, dass alle Wortkombinationen überhaupt, bzw. in ausreichender Menge für eine verlässliche statistische Schätzung beinhaltet sind.

Der zweite Term in Gl. 1.2 beschreibt die Likelihood der Musterfolge $\boldsymbol{\mathcal{X}}$ bei gegebener Wortfolge $\boldsymbol{w}$. Da die eigentliche Wahrscheinlichkeit $p(\boldsymbol{\mathcal{X}}|\boldsymbol{w})$ jedoch nicht bekannt ist, wird versucht, diese durch ein stochastisches, parametrisches Modell $p_\Lambda(\boldsymbol{\mathcal{X}}|\boldsymbol{w})$ nachzubilden. Bewährt haben sich hierzu die sogenannten Hidden-Markov-Modelle (HMM) [Rab86, Rab89], in deren

Zuständen, meist mittels Gauss'scher Normalverteilungen, die eigentliche Wahrscheinlichkeitsdichtefunktion (WDF) approximiert wird.

1.2.2 Stochastische Modellierung mit Hidden-Markov-Modellen

Abb. 1.2 zeigt die Struktur eines HMMs mit 3 Zuständen das ausschließlich von links nach rechts abgearbeitet werden kann.

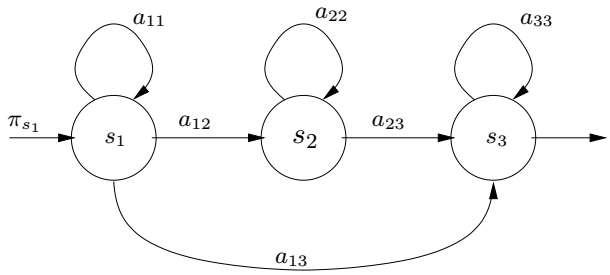

Abb. 1.2: Hidden-Markov-Modell mit 3 Zuständen.

π_{s_1} beschreibt die Einsprungwahrscheinlichkeit in den ersten Zustand des HMMs. Sie ergibt sich zu $\pi_{s_1} = 1.0$, falls ausschließlich dieser Einsprung in das Modell gestattet ist. Die Parameter $a_{i,i}$ geben die Wahrscheinlichkeit für einen Verbleib im jeweiligen Zustand i. Der Verbleib wird auch als Schleife (engl. 'loop') bezeichnet. Die Wahrscheinlichkeit in den nächsten Zustand zu wechseln (engl. 'next') ist durch $a_{i,i+1}$ gegeben. In praktischen Systemen, insbesondere für eine optimierte Erfassung der Sprechgeschwindigkeit, werden häufig noch Übergänge (engl. 'skip'), unter Auslassung eines Zustands, zugelassen. Die Wahrscheinlichkeit für eine Auslassung wird durch $a_{i,i+2}$ beschrieben. Die Einteilung eines HMMs in Einzelzustände hat zum Ziel, die Zeitstruktur einer Spracheinheit - etwa Anfang, Mitte und Ende - zu erfassen. Jeder Zustand soll einen möglichst stationären Bereich des Musterverlaufs repräsentieren. In den Einzelzuständen wird die Emissionswahrscheinlichkeitsdichte $p(\boldsymbol{x}_j|m,s)$ berechnet, die angibt, wie wahrscheinlich dieser Zustand des Modells in der Lage ist, den gegebenen Mustervektor zu erzeugen. Die Teil-WDF innerhalb eines Zustands s eines Modells m wird durch eine gewichtete Überlagerung Gauss'scher Normalverteilungen (Gl. 1.4) realisiert.

$$p(\boldsymbol{x}_j|m,s) \approx p_\Lambda(\boldsymbol{x}_j|m,s) = \sum_{k=1}^{K_{ms}} c_{msk}\mathcal{N}(\boldsymbol{x}_j, \boldsymbol{\mu}_{msk}, \boldsymbol{\Sigma}_{msk}) \tag{1.4}$$

Die Mixturkoeffizienten c_{msk} haben die 2 probabilistischen Nebenbedingungen zu erfüllen:

$$c_{msk} > 0 \quad \forall k \quad \text{und} \quad \sum_{k=1}^{K_{ms}} c_{msk} = 1 \tag{1.5}$$

Bei der Wahl der zu modellierenden sprachlichen Einheiten reicht das Spektrum von Worteinheiten, über Silben, Halbsilben [Rus94], bis zu Phonemen. Bei den meisten heutigen

Systemen hat es sich durchgesetzt Phoneme, bzw. Phoneme in Abhängigkeit ihres Lautkontexts [Bah91] zu modellieren. Ein jedes Phonem wird hierbei durch ein eigenes HMM (oder mehrere - bei kontextabhängiger Modellierung) repräsentiert. Aus diesen wiederum lässt sich durch Konkatenation (s. Abb. 1.2) der HMMs, basierend auf einer lautsprachlichen Umschrift, eine vollständige Worteinheit generieren. Die Art und Weise, wie die Lautmodelle zu Worteinheiten zu verketten sind, wird einem Aussprachelexikon (vgl. Abb. 1.4) entnommen. Dieses orientiert sich im Deutschen häufig an der Dudenumschrift [Dud90]. Da die Aussprache jedoch, gerade bei Spontansprache, oft von der Dudennorm abweicht, wird es gerne durch gebräuchliche Aussprachevarianten [Pfa00b, Fin97b] ergänzt.

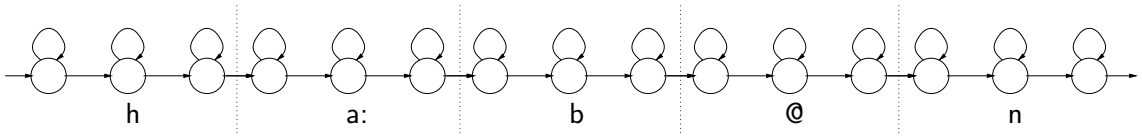

Abb. 1.3: Konkatenation von Phonem-HMMs mit jeweils 3 Zuständen zu einem kompletten Wortmodell für das Wort "haben" (in Sampa Notation).

Die Parameter der Hidden-Markov-Modelle, d.h. insbesondere die Parameter der zugrundeliegenden Mixturverteilungen (z.B. Gl. 1.4), werden aus großen Sprachstichproben geschätzt. Die Verfahren, die hierbei zur Anwendung kommen, sind Thema des Kapitels 2.

1.2.3 Viterbi-Algorithmus

Die Gesamtwahrscheinlichkeit einer Vektorfolge bezüglich eines HMMs berechnet sich aus der Summation über alle Pfadwahrscheinlichkeiten, d.h. die Summe über die Wahrscheinlichkeiten aller theoretisch möglichen Wege $\boldsymbol{s} = \{s_1, s_2, \ldots, s_T\}$ durch das HMM. Die Pfadwahrscheinlichkeit $p(\boldsymbol{\mathcal{X}}|m, \boldsymbol{s})$ eines Einzelpfads folgt aus dem Produkt der Einzelwahrscheinlichkeiten (Gl. 1.6).

$$p(\boldsymbol{\mathcal{X}}|m, \boldsymbol{s}) = \pi_{s_1}\{\prod_{t=1}^{T-1} a_{s_t s_{t+1}} p(\boldsymbol{x}_t|s_t)\} p(\boldsymbol{x}_T|s_T) \tag{1.6}$$

$$p(\boldsymbol{\mathcal{X}}|m) = \sum_{\Theta_{\boldsymbol{s}}} p(\boldsymbol{\mathcal{X}}|m, \boldsymbol{s}) \tag{1.7}$$

Die Berechnung der Wahrscheinlichkeit $p(\boldsymbol{\mathcal{X}}|m)$ kann rekursiv mit dem sog. Forward-Backward-Algorithmus [Rab89, Rus94] erfolgen. Eine Approximation von $p(\boldsymbol{\mathcal{X}}|m)$ ergibt sich, wenn statt aller möglichen Pfade $\Theta_{\boldsymbol{s}}$ durch das HMM, nur derjenige in Betracht gezogen wird, der die höchste Wahrscheinlichkeit aufweist, d.h. $p(\boldsymbol{\mathcal{X}}|m) \approx p(\boldsymbol{\mathcal{X}}|m, \boldsymbol{s}^*)$ mit

$$p(\boldsymbol{\mathcal{X}}|m, \boldsymbol{s}^*) = \max_{\boldsymbol{s}} \pi_{s_1} \left[\prod_{t=1}^{T-1} a_{s_t s_{t+1}} p(\boldsymbol{x}_t|s_t)\right] p(\boldsymbol{x}_T|s_T) = \tag{1.8}$$

$$= \max_{\boldsymbol{s}} \left[\log \pi_{s_1} + \left[\sum_{t=1}^{T-1} (\log a_{s_t s_{t+1}} + \log p(\boldsymbol{x}_t|s_t))\right] + \log p(\boldsymbol{x}_T|s_T)\right] \tag{1.9}$$

Die Berechnung von $p(\boldsymbol{\mathcal{X}}|m, \boldsymbol{s}^*)$ erfolgt mit dem sogenannten Viterbi-Algorithmus [For73]. Dieser Algorithmus ermöglicht zugleich die Bestimmung der Segmentgrenzen, falls die Berechnung über verkettete HMMs (s. Abb. 1.3) hinweg erfolgt. Dieser Ansatz ist Bestandteil des Segmental K-Means Trainingsverfahrens [Jua90], das zum Schätzen der Modellparameter eingesetzt wird und das im Abschnitt 2.3.1 diskutiert wird.

1.3 Basissystem

Das verwendete Basissystem ist ein Spracherkennungssystem für kontinuierliche, spontansprachliche Äußerungen [Beh95a, Pla95]. Abb. 1.4 zeigt schematisch den Aufbau des Systems und die integrierten Wissensquellen.

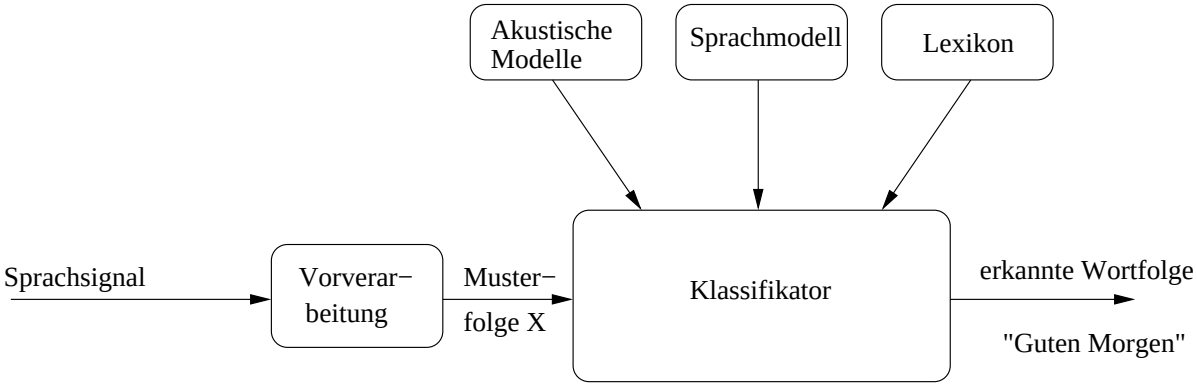

Abb. 1.4: Aufbau des Basissytems mit den integrierten Wissensquellen.

Der Erkenner ist für einen mittleren Wortschatz von ca. 5000 Wörtern konzipiert und benutzt ein Bigramm-Sprachmodell. Die eingesetzten Hidden-Markov-Modelle können als einfache Monophone, oder als Triphone mit rechts- und linksseitiger Kontextmodellierung (s. Kap. 4.1) ausgelegt werden.

1.4 Vorverarbeitung und Merkmalsbildung

Die Vorverarbeitung (s. Abb. 1.4) bereitet aus dem akustischen Zeitsignal eine Folge von zeitdiskreten Mustervektoren auf. Die Mustervektoren setzen auf einer Transformation in den Spektralbereich auf. Aufgrund des instationären Verhaltens der menschlichen Sprache muss dabei eine Kurzzeitspektralanalyse zugrundegelegt werden. Innerhalb des kurzen Berechnungszeitraums wird Quasi-Stationärität angenommen. Im vorliegenden System [Beh95a] werden hierzu folgende Einstellungen verwendet:

Abtastfrequenz:	16kHz
Quantisierung:	16bit
Fenster:	Hamming
Fensterbreite:	16ms (256 Samplepunkte)
Fenstervorschub:	10ms

Die Fenster überlappen um 6ms. Das mit 16kHz abgetastete und mit 16bit quantisierte Zeitsignal wird durch die Vorverarbeitung in eine Folge von Mustervektoren im Abstand von 10ms transformiert. Jedes Fenster, auch als Rahmen (engl.: 'Frame') bezeichnet, wird dazu mittels einer schnellen Fourier-Transformation (FFT) in den Spektralbereich transformiert. Aus dem FFT-Spektrum wird nach Kanalbildung und einer gehörorientierten Weiterverarbeitung der letztendliche Merkmalsvektor gebildet. Zwei Arten der Weiterverarbeitung wurden primär in den Experimenten verwendet. Die bestehende, lautheitsbasierte Münchner Vorverarbeitung [Beh95a] wurde hierzu durch eine mel-cepstrale Weiterverarbeitung ergänzt.

1.4.1 Lautheitsbasierte Vorverarbeitung

Die im folgenden als MUC bezeichnete "Münchner" Vorverarbeitung orientiert sich an der psychoakustischen Empfindungsgröße Lautheit, die angibt, wie laut ein Schallereignis vom Menschen empfunden wird [Vog75, Rus94, Beh95a]. Die Basis ist eine Zusammenfassung des Spektrums zu Frequenzgruppen anhand der nichtlinearen Bark-Skala, wobei 20 Lautheitskanäle im Abstand von 1 Bark verwendet werden. In den endgültigen Merkmalsvektor werden neben den 20 spektralen Kanälen, deren 1. und 2. Ableitungen noch die normierte Gesamtlautheit, die Nulldurchgangsrate, sowie die 1. und 2. Ableitung der modifizierten Lautheit und der Gesamtlautheit aufgenommen [Rus94, Beh95a, Pla95, Pfa00b].

1.4.2 Mel-cepstrale Vorverarbeitung

Analog zur Münchner handelt es sich auch bei der mel-cepstralen-Vorverarbeitung um einen gehörorienten Ansatz. Die Kanaleinteilung erfolgt hier anhand der Mel-Skala [Dav80], die einen logarithmischen Verlauf aufweist und im unteren Bereich, bis ca. 1kHz, nahezu linear verläuft.

$$f_{mel}(f) = 2595 \log(1 + \frac{f}{700Hz}) \tag{1.10}$$

Die Kanaleinteilung auf dieser Skala ist jedoch nicht fest vorgegeben. Vorgeschlagen wurden hier Einteilungen im Bereich von 17 bis 25 Frequenzgruppen. Die Kanalfilterung erfolgt mittels asymmetrischer Dreiecksfilter (Sägezahn). In der implementierten Variante wurde der Vorschlag nach [ChR97] aufgegriffen, der mit 21 Kanälen arbeitet. Die Breite der Filter nimmt oberhalb von 1kHz mit steigender Frequenz weiter zu. Auf den einzelnen Mel-Kanälen folgt eine Logarithmierung der enthaltenen Leistung mit anschließender spektraler Rücktransformation. Als Spektraltransformation wird eine Diskrete Kosinus-Transformation (DCT) eingesetzt, bei der die obersten Cepstral-Koeffizienten verworfen werden. Die Quasi-Rücktransformation in den Zeitbereich ist in dieser Form auch unter dem Begriff Cepstraltransformation bekannt. Die einzelnen Koeffizienten werden daher als MFCCs (engl.: 'Mel Frequency Cepstral Coefficients') bezeichnet.

1.4.3 Weitere Merkmale

Neben den rein "spektralen" Merkmalen werden in den Mustervektor häufig noch eine Reihe zusätzlicher Merkmale hinzugenommen. Zu den wichtigsten, auch in dieser Arbeit, zählen

insbesondere die Gesamtenergie und die Nulldurchgangsrate (NDR). Charakteristisch ist eine hohe Gesamtenergie insbesondere für stimmhafte Sprachabschnitte. Niedrige Gesamtenergie findet sich bei stimmlosen Frikativen sowie Sprech- und Verschlusspausen. Ähnlich charakteristisch ist für solche Abschnitte die sogenannte Nulldurchgangsrate, die die Anzahl der Vorzeichenwechsel des Zeitsignals innerhalb des Fensters angibt. Aus ihr lässt sich ablesen, ob die Anregung in einem Sprachabschnitt stimmhaft oder -los ist. Eine hohe NDR ergibt sich entsprechend für stimmlose Frikative, wohingegen Nasale, Gleitlaute und Vokale eher geringe NDR zeigen [Beh95a].

1.4.4 Delta-Koeffizienten

In den Zuständen eines HMMs wird, gemäß Gl. 1.4, die Emissionswahrscheinlichkeitsdichte eines Vektors bestimmt, d.h. im HMM selbst werden keine Abhängigkeiten zwischen den Vektoren berücksichtigt. Die aufeinanderfolgenden Mustervektoren eines Sprachsignals sind allerdings hochgradig korreliert. Um diesem Defizit Rechnung zu tragen, werden in der Vorverarbeitung die sog. Delta-Koeffizienten (Δ) berechnet und dem Mustervektor hinzugefügt. Diese Ableitungskoeffizienten geben die Änderung zwischen Vektoren wieder. Berechnet werden sie bei zeitdiskreten Werten anhand einer Differenzengleichung

$$d_1(k) = \sum_{n=-\Delta}^{\Delta} \frac{n}{\Delta} x(n+k) \tag{1.11}$$

wobei häufig $\Delta = 2$ ist. Ergänzt werden die Delta-Koeffizienten meist durch Beschleunigungskoeffizienten (DeltaDelta,$\Delta\Delta$), die die Änderung der Ableitungskoeffizienten erfassen. Gebildet werden sie im vorliegenden System durch

$$d_2(k) = \sum_{n=-\Delta}^{\Delta} \frac{n}{\Delta} d_1(n+k) \tag{1.12}$$

1.4.5 Nomenklatur der Vorverarbeitungen

In dieser Arbeit werden die beiden Vorverarbeitungen MUC und MFCC in verschiedenen Konstellationen untersucht und eingesetzt - meist in Kombination mit erweiterten Merkmalen sowie Delta(Delta)-Koeffizienten. Die im folgenden verwendete Bezeichnung richtet sich dabei nach der Basisvorverarbeitung und der Gesamtzahl der in den Merkmalsvektor aufgenommen Koeffizienten, einschließlich der Delta- und der erweiterten Koeffizienten.

MUC20	nur 20 Lautheitskanäle
MUC66	(20 Lautheiten) + 20 Deltas + 20 DeltaDeltas + norm. Gesamtlautheit + NDR + (mod. Lautheit + Gesamtlautheit) + 2 Deltas + 2 DeltaDeltas
MFCC12	nur 12 Mel-Cepstren
MFCC24	(12 Mel-Cepstren) + 12 Deltas
MFCC36	(12 Mel-Cepstren) + 12 Deltas + 12 DeltaDeltas
MFCC42	(12 Mel-Cepstren + Energie + NDR) + 14 Deltas + 14 DeltaDeltas

Die beiden beiden als MUC66 und MFCC42 bezeichneten Vorverarbeitungen stellen die Grundlage des vorgestellten Spracherkennungssystems dar. Die übrigen kommen vorzugsweise für Sprechererkennungs- und zuordnungsaufgaben zum Einsatz. Speziell die Zusammensetzung MFCC12 hat hierbei aufgrund der geringen Größe den Vorteil des verringerten Rechenaufwands auf ihrer Seite.

1.5 Verwendete Datenbasis

Die Grundlage der durchgeführten Untersuchungen und Experimente bildet der VERBMOBIL Korpus. Die aufgezeichneten Dialoge dieser Sprachdatensammlung entstammen der Domäne 'Terminvereinbarung', bei der jeweils zwei Gesprächspartner versuchen sich auf einen gemeinsamen Termin zu verständigen. Die Daten können als spontansprachlich angesehen werden und sind daher hochgradig vom Sprechstil beeinflusst.

Die Trainingsdatenmenge umfasst die CDs 1-5,7 und 12 mit insgesamt 11355 Äußerungen. Die Daten wurden in Karlsruhe, Kiel, Bonn und München aufgenommen und stammen von insgesamt 613 Sprechern. Von diesen sind 332 männlich, 281 weiblich. Als Testkorpus für Spracherkennungsexperimente wurde das offizielle Eval96-Testset [ReJ96], das 343 Äußerungen von 30 Sprechern umfasst, eingesetzt. Das Testlexikon besteht aus 5193 Wörtern der genannten Domäne. Zusätzlich stand ein Bigramm-Sprachmodell der Fa. Philips zur Verfügung.

Für die Bewertung der Sprechererkennungs- bzw. -zuordnungsaufgaben wurde das Crossvalidierungsset Xval96 mit 599 Turns (=Äußerungen) zur Basis genommen. Von diesem wurden jedoch diejenigen Sprecher ausgenommen, die nicht in den Trainingsdaten enthalten sind. Es verbleiben 396 Turns von 74 Sprechern.

Für vergleichende Adaptionsexperimente und zur Bewertung der Adaptionsalgorithmen wurden 40 Sprecher von VERBMOBIL CD14 ausgewählt. Von jedem Sprecher standen 2 bzw. 10 Turns zur Systemanpassung zur Verfügung. Das zugehörige Testset beinhaltet insgesamt 524 Testsätze der gewählten 40 Sprecher.

Kapitel 2

Initialisierung und Training akustischer Modelle

An dieser Stelle erfolgt eine zentrale Diskussion der in dieser Arbeit untersuchten und eingesetzten Verfahren zur Parameterschätzung. Darüber hinaus werden einige grundlegende Gruppierungsprinzipien vorgestellt. Auf diese Verfahren wird in den daran anschließenden Kapiteln zurückgegriffen.

2.1 Einführung

Wie im Einführungsteil dieser Arbeit beschrieben, stellt sich bei der Erkennung gesprochener Sprache das in Gl. 1.2 angegebene Dekodierungsproblem. In dieser Gleichung wird die Likelihood $p(\boldsymbol{\mathcal{X}}|\boldsymbol{w})$ für beliebige Wortfolgen $\boldsymbol{w}$ benötigt. Im Rahmen der stochastischen Modellbildung wird ein Wortmodell durch die Konkatenation von einzelnen, meist phonemweisen HMM-Modellen erzeugt (s. Abb. 1.3). In den Zuständen dieser HMMs wird mittels überlagerter Normalverteilungen die reale WDF $p(\boldsymbol{x}|m,s)$ eines Lauts m im Merkmalsraum parametrisiert nachgebildet (Gl. 1.4). Für eine verlässliche statistische Repräsentation $p_\Lambda(\boldsymbol{x}|m,s)$ müssen dazu aus vorhandenen Sprachdaten die Kenngrößen der Prototypverteilungen geschätzt werden. Bei Gauss'schen Normalverteilungen sind dies die Mittelpunkte sowie die Kovarianzmatrizen.

Bei der parametrisierten Nachbildung der realen Verteilung der Merkmalsvektoren ergeben sich 2 Problemstellungen. Für jeden Zustand eines jeden HMMs muss eine *geeignete Anzahl* an zu überlagernden Prototypverteilungen (z.B. Mittelpunktsvektoren, Normalverteilungen, Laplaceverteilungen) festgelegt werden, für die dann aus einer Trainingsstichprobe die beschreibenden Parameter (Mittelpunkte, Kovarianzen) geschätzt werden müssen. Für diesen zweiten Schritt wurden in der Literatur eine Reihe sogenannter Trainingsverfahren vorgestellt. Diese werden ab Abschnitt 2.3 diskutiert. Die Festlegung einer optimalen Anzahl von Verteilungen zur Repräsentation einer gegebenen Verteilung von Mustervektoren ist ein nicht-triviales Problem, für das in der Literatur noch keine befriedigende, allgemeingültige Lösung existiert. Im folgenden Abschnitt werden einige Ansätze diskutiert, die dieser Aufgabenstellung mit iterativen Clusterverfahren begegnen.

Die zentrale, in dieser Arbeit untersuchte Aufgabenstellung ist die Integration von Kontextwissen (Sprecher, Sprechgeschwindigkeit) durch explizite Gruppenbildungen. Die angeführten Initialisierungsverfahren stellen die unterste Ebene der Gruppierung in einem Spracherkennungssystem dar, bei der Trainingsvektoren zu Gruppen zusammengefasst werden. Das Ziel ist hierbei, innerhalb der vorgegebenen Mustervektoren Ballungen (engl. 'Cluster') ausfindig zu machen, die durch einen parametrischen Prototypen repräsentiert werden können. In diesem Zusammenhang wird mit dem OC-Clusterverfahren ein Ansatz vorgestellt, der die Anpassung der Verteilungszahl für spezifische Sprechgeschwindigkeiten zum Ziel hat. Die Problemstellung, eine geeignete Modellgröße festzulegen, ist eng mit der Thematik der Modellinitialisierung verknüpft. Um mittels eines iterativen Trainingsverfahrens zu einer möglichst optimalen Parameterschätzung zu gelangen, sollten die Ausgangswerte(=Initialisierung) der Parameter beim Start des Trainingsalgorithmus bereits nahe am Optimum liegen [Rus94].

2.2 Initialisierungsverfahren

2.2.1 Stand der Technik

Bei der Initialisierung sowohl von HMM-Zuständen, als auch von sprecherspezifischen GMM-Modellen (s. Kap. 5.2.1.2), hat sich der Einsatz sogenannter "Top-Down"-Clusteralgorithmen durchgesetzt. Die zugrundeliegende Idee ist durch eine fortgesetzte Erhöhung der Prototypenzahl die Modellierung an die vorhandene Trainingsdatenmenge anzupassen. Das Ziel ist, Vektorballungen (Cluster) im Merkmalsraum zu finden, die durch einen Prototypen repräsentiert werden können. Eines der bekanntesten Verfahren, das auch als direkter Trainingsalgorithmus für VQ-Modelle (s. Kap. 5.2.1.1) eingesetzt wird, ist das Verfahren nach Linde-Buzo-Gray (LBG) [Lin80, Wol97]. Im Prinzip handelt es sich bei diesem Ansatz um eine Kombination aus einer inkrementellen Erhöhung der Prototypenzahl und einem K-Means Algorithmus zur Optimierung der Zuordnung zwischen Mustervektoren und Lage der Prototypen.

Startpunkt des LBG-Verfahrens ist der globale Mittelpunkt der zu modellierenden Daten. Dieser wird in 2 Prototypen aufgespalten. Die Teilung erfolgt - bei Mittelpunktsprototypen mit diagonaler Kovarianzmatrix - aus Gründen der Einfachheit und Schnelligkeit meist in Richtung des Varianzvektors. Im Anschluss daran werden die Mustervektoren anhand des Euklidischen Abstands den Prototypen neu hart zugeordnet. Jeder Prototyp wird nun als Mittelpunkt seiner Musterteilmenge neu berechnet. Die Abfolge von Neuzuordnung der Vektoren und Neuberechnung der Prototypen wird solange wiederholt, bis keine Verbesserung des Störungsmaßes erzielt werden kann. Indem in der nächsten Iteration i nun diese beiden Prototypen geteilt werden, beginnt der Vorgang von neuem. Die Prototypenzahl verdoppelt sich bei jeder Teilungsoperation, was den Algorithmus sehr schnell macht. Der Algorithmus wird fortgesetzt, bis die gewünschte Zahl an Prototypen erreicht wird. Als Störungsmaß wird vorzugsweise der mittlere quadratische Fehler $D_{MSE}(i)$ (MSE, engl. 'Mean Square Error') verwendet.

$$D_{MSE}(i) = \frac{1}{\sum_{k=1}^{K(i)} N_P^k} \sum_{k=1}^{K(i)} \left(\sum_{j=1}^{N_P^k} |\boldsymbol{x}_{kj} - \boldsymbol{\mu}_k|^2 \right) \tag{2.1}$$

Der iterative Bestandteil des LBG-Algorithmus, bestehend aus der abwechselnden Neuzuordnung der Vektoren und der Neuberechnung der Prototypen (bei konstanter Prototypenzahl K), ist auch als K-Means Algorithmus bekannt. Er kann auch als separates Verfahren zur Initialisierung eingesetzt werden. In diesem Fall können, um die K nötigen Prototypen zu erhalten, einfach K zufällig ausgewählte Mustervektoren als initiale Prototypen erklärt werden. Dem Vorteil einer deutlichen Reduktion des Rechenaufwands steht jedoch der Nachteil einer verringerten Genauigkeit der Repräsentation gegenüber.

2.2.2 Selektive Clusterteilung

Eine Modifikation des LBG-Verfahrens ergibt sich durch eine selektive Auswahl der Prototypen, die in einer Iteration zu teilen sind. In dieser Arbeit wurden verschiedene Auswahlkriterien untersucht, anhand derer die Prototypselektion durchgeführt werden kann, u.a. (Pseudo)Mixturgewicht, Clustervarianz sowie Anzahl der Vektoren in einem Cluster. Als sehr effektives Kriterium (s. Kap. 5.2.3) hat sich das Emissions- oder Likelihoodverhältnis (EMR, engl. "EMission Ratio") [Wol97] zweier Vektorcluster gezeigt.

$$EMR = \frac{1}{N_P^{12}} \log \frac{p_1(\boldsymbol{\mathcal{X}}_1) p_2(\boldsymbol{\mathcal{X}}_2)}{p_{12}(\boldsymbol{\mathcal{X}}_{12})} \qquad \text{mit} \qquad \boldsymbol{\mathcal{X}}_{12} = \boldsymbol{\mathcal{X}}_1 \cup \boldsymbol{\mathcal{X}}_2 \tag{2.2}$$

In einer Iteration wird nur derjenige der K Prototypen gespalten, dessen Teilung den höchsten Likelihood-Gewinn für die, dem Cluster zugewiesenen Mustervektoren $\boldsymbol{\mathcal{X}}_{12}$ verspricht. Eine weitergehende Optimierung konnte durch die Verwendung von Crossvalidierungsdaten [Kem95, Fal99] erreicht werden. Dieses Vorgehen wird im folgenden Abschnitt 2.2.3 beschrieben.

2.2.3 Optimierte Vektorgruppierung

Die bei den bisherigen Betrachtungen noch nicht berücksichtigte Problemstellung ist die Festlegung der Anzahl an benötigten Prototypen. Bei den iterativen Top-Down Clusterverfahren ist diese Frage gleichzusetzen mit der Festlegung eines Abbruchkriteriums für den Teilungsprozess. Ein einfacher Weg, dieser Fragestellung zu begegnen, ist, neben der Festlegung einer fixen Prototypzahl, die Berücksichtigung der Zahl der Vektoren, die einem Prototypen $\boldsymbol{\mu}_i$ hart zugeordnet werden. Dies geschieht meist in Form eines Schwellwerts N_P^{min} [Wol97], unterhalb dessen ein Prototyp nicht mehr geteilt werden darf.

Die Forderung einer Mindestzahl an Vektoren je Prototyp ist zwar einerseits nötig, um eine robuste Schätzung zu gewährleisten, andererseits ist diese Festlegung jedoch u.U. sehr grob, da keine Aussagen über die eigentliche Verteilung der Vektoren beinhaltet sind. Im Rahmen dieser Arbeit wurde daher ein Algorithmus [Fal99] (OC, engl. "Optimized Clustering") entwickelt und implementiert, mit der Maßgabe, die Zahl der Prototypen für gegebene Trainingsdaten zu optimieren. Als Optimierungskriterium wurde dazu die Generalisierungsfähigkeit der Prototypenverteilung herangezogen. Unter Generalisierung ist die Fähigkeit der erzeugten Prototypen zu verstehen neue und unbekannte Daten zu repräsentieren. Realisiert wurde dies durch eine Teilung der verfügbaren Merkmalsvektoren in eine reine Trainingsmenge und eine

Crossvalidierungsmenge (CV). Die Generalisierungsfähigkeit kann anhand der erzielten Likelihood der CV-Daten bezüglich der gegebenen Prototypen abgeschätzt werden. Ein solches Vorgehen wurde auch schon von Kemp [Kem95] vorgeschlagen, der die CV-Daten jedoch nur zur Bewertung der globalen Performanz heranzog. Die Crossvalidierungsdaten dienen dazu, die "unbekannten" Testdaten zu simulieren. Allerdings können die abgespaltenen CV-Daten nur bedingt als Simulation unbekannter Daten gesehen werden, da die intrinsische Ähnlichkeit der Trainingsdaten meist hoch ist und sie somit unbekannte, abweichende Daten nur ungenügend repräsentieren. Darüber hinaus muss bei der Teilung der Trainingsdaten sehr vorsichtig vorgegangen werden, da die Zweiteilung eine Reduktion der verfügbaren reinen Trainingsdatenmenge bedingt. Diese Verminderung erschwert die gewünschte robuste Parameterschätzung. Im entwickelten Algorithmus wird diesem Problem Rechnung getragen, in dem die CV-Daten zusätzlich zur Auswahl der zu teilenden Prototypen herangezogen werden. Dadurch werden sie zwar nicht direkt zur Parameterschätzung verwendet, sind andererseits jedoch der Modellierung auch nicht vollends entzogen. Der Ablauf des Verfahrens lässt sich wie folgt strukturieren:

```
Teile vorhandene Daten in Trainings- und
        Crossvalidierungsdaten (CV)
Berechne globalen Mittelpunkt der Trainingsdaten

wiederhole

     Maximumsuche:
     berechne theoretischen Likelihood-Gewinn bei Teilung
          eines jeden der K Prototypen auf den CV-Daten

     teile den Prototypen mit maximalem Gewinn,
          falls Gewinn > Gewinnschwellwert
          sonst Ende

     Iterative Neuzuordnung der Trainingsdaten bzw.
          Neuberechnung K -> K+1

solange K<Kmax
```

Der als "Maximumsuche" bezeichnete Teilblock in obigem Schema zielt darauf ab den Prototypen zu finden, dessen Teilung den höchsten Gewinn erbringt. Dazu wird für jeden Prototypen eine "simulierte" Teilung einschließlich K-Means Optimierung durchgeführt. Um eine möglichst genaue Teilung zu erzielen, wurde für jeden Prototypen die Hauptstreurichtung der ihm zugeordneten Vektoren bestimmt. Die Hauptstreurichtung ergibt sich als Eigenvektor $\boldsymbol{e}_{\lambda_1}$ zum größten Eigenwert λ_1 der Kovarianzmatrix der Vektorgruppe. Sie ist gleichbedeutend mit der ersten Hauptachse einer Hauptachsentransformation (PCA, engl. 'Principle Components Analysis'). Die beiden neuen, verschobenen Prototypen (vor Neuzuordnung) ergeben sich zu:

$$\boldsymbol{\mu}_k^{1,2} = \boldsymbol{\mu}_k \pm 0.2\sigma_{\lambda_1}\boldsymbol{e}_{\lambda_1} \tag{2.3}$$

Die Verschiebung in Richtung der Hauptstreurichtung erfolgt gewichtet mit der Standardabweichung der Vektorverteilung in dieser Richtung. Um andere Mittelpunkte wenig zu beeinflussen, wird die Verschiebung mit einem Skalierungsterm ≈ 0.2 beaufschlagt. Diese Teilung reicht aus um die Vektoren aufzuteilen und sie für die iterative K-Means Neuzuordnung vorzubereiten. Jede der simulierten Teilungen k (von K Prototypen) wird anhand des Likelihoodgewinns auf den CV-Daten bewertet.

$$LL_K = \sum_{j=1}^{N_P^{CV}} \log \sum_{k=1}^{K} c_k \mathcal{N}(\boldsymbol{x}_j, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{2.4}$$

$$LL_{K+1}(k) = \sum_{j=1}^{N_P^{CV}} \log \sum_{k^*=1}^{K+1} c_{k^*} \mathcal{N}(\boldsymbol{x}_j, \boldsymbol{\mu}_{k^*}, \boldsymbol{\Sigma}_{k^*}) \tag{2.5}$$

$$\hat{k} = \arg\max_k \Delta LL(k) = \arg\max_k (LL_{K+1}(k) - LL_K) \tag{2.6}$$

Nach der Auswertung der K simulierten Teilungen wird diejenige Teilung tatsächlich ausgeführt, die den höchsten Likelihoodzuwachs verspricht. Da sich Trainings- und Crossvalidierungsdaten unterscheiden, ist ab einer gewissen Zahl von Prototypen kein Zuwachs an Likelihood mehr zu erwarten und der Algorithmus terminiert automatisch.

Das beschriebene, und in [Fal99] erstmals vorgestellte OC-Clusterverfahren wurde für eine, bzgl. der Sprechgeschwindigkeit, optimierte Initialisierung konzipiert. Prinzipiell ist dieses Verfahren jedoch auch für allgemeinere Vektorgruppierungsprobleme geeignet. Mit gutem Erfolg konnte es auch zur automatischen Initialisierung bzw. zum Training sprecherspezifischer GMM-Modelle [Rey95] eingesetzt werden (s. Kap. 5.2.1.2). Vorteil dieser Methode ist die automatische Anpassung der Prototypenzahl an die zu modellierende Vektorverteilung. Die Festlegung hängt also nicht mehr vorwiegend vom Expertenwissen des Systemdesigners ab. Ein weiterer Vorteil ergibt sich durch die Optimierung der Prototypen mit Bezug auf unbekannte Daten - simuliert durch Crossvalidierungsdaten. Die Generalisierungsfähigkeit wird dadurch stark erhöht. Bei Sprechererkennungsexperimenten hat sich gerade dieser Ansatz als sehr effektiv erwiesen (s. Abschnitt 5.2.3). Als deutlicher Nachteil des entwickelten Verfahrens muss der stark erhöhte Rechenaufwand angeführt werden. Der Mehraufwand im Vergleich zu LBG wird insbesondere durch die Einführung der PCA - zur Berechnung der optimalen Teilungsrichtung - erzeugt. Verstärkt wird dies zusätzlich durch das Testen der Prototypen mittels einer simulierten Teilung.

2.3 Grundlegende Trainings- und Adaptionsverfahren

2.3.1 Der "Segmental K-Means"-Ansatz

Die im folgenden diskutierten Trainingsverfahren werden für den allgemeineren Fall einer Folge von HMM-Modellen abgeleitet. Die Gleichungen sind ohne Einschränkung auch für

Gauss'sche Mixtur-Modelle anwendbar, da es sich bei diesen de-facto um ein HMM - bestehend aus nur einem einzelnen Zustand - handelt. Für das Training einer Folge von HMM-Modellen anhand des Maximum-Likelihood-Prinzips sind insbesondere zwei Techniken bekannt geworden. Beim Segmental K-Means (auch Viterbi-) Training, dessen Konvergenz in [Jua90] gezeigt wurde, erfolgt die Optimierung der Modellparameter nur anhand der (optimalen) Zustandssequenz $\boldsymbol{s}^* = \{s_1^*, s_2^*, \ldots, s_T^*\}$, welche die maximale Wahrscheinlichkeit aufweist. Ein Vektor beeinflusst daher nur die Verteilungsparameter des einen, anhand des Viterbi-Pfades zugewiesenen Zustands. Im Gegensatz hierzu wird beim Forward-Backward (auch Baum-Welch-) Training [Rab86, Rab89, Rus94] eine weiche, wahrscheinlichkeitsorientierte Zuordnung zu den Zuständen vorgenommen, die entsprechend alle Pfade durch ein HMM berücksichtigt. Die Diskussion im folgenden konzentriert sich auf das in dieser Arbeit implementierte und verwendete Segmental K-Means Prinzip. Das zugrundeliegende Zielkriterium ergibt sich nach [Jua90] zu:

$$\max_{\mathbf{s}} p(\boldsymbol{\mathcal{X}}, \boldsymbol{s}|\lambda) = \max_{\mathbf{s}} \pi_{s_1} \prod_{t=1}^{T} a_{s_{t-1}s_t} b_{s_t}(\boldsymbol{x}_t) \tag{2.7}$$

$$\begin{aligned} \bar{\lambda} &= \arg\max_{\lambda} \Big[\max_{\mathbf{s}} p(\boldsymbol{\mathcal{X}}, \boldsymbol{s}|\lambda)\Big] = \\ &= \arg\max_{\lambda} \Big[\max_{\mathbf{s}} \left[\log p(\boldsymbol{\mathcal{X}}|\boldsymbol{s}, \lambda) + \log p(\boldsymbol{s}|\lambda)\right]\Big] \end{aligned} \tag{2.8}$$

Juang [Jua90] konnte zeigen, dass die Maximierung dieser Zielfunktion in zwei Teilschritte zerfällt. Im ersten Schritt kann mittels des Viterbi-Algorithmus die Bestimmung des optimalen Zustandspfads $\boldsymbol{s}^*$ erfolgen:

$$\boldsymbol{s}^* = \arg\max_{\mathbf{s}} p(\boldsymbol{\mathcal{X}}, \boldsymbol{s}|\lambda) \tag{2.9}$$

Die nachfolgende Parametermaximierung lässt sich, wie Gl. 2.8 zeigt, wiederum in zwei Terme zergliedern, die unabhängig voneinander optimiert werden können.

$$\log p(\boldsymbol{\mathcal{X}}|\boldsymbol{s}^*, \lambda) = \sum_{t=1}^{T} \log b_{s_t^*}(\boldsymbol{x}_t) \tag{2.10}$$

$$\log p(\boldsymbol{s}^*|\lambda) = \sum_{t=1}^{T} \log a_{s_{t-1}^* s_t^*} \tag{2.11}$$

Die Optimierung von reinen Verteilungsparametern reduziert sich somit auf die Betrachtung von $p(\boldsymbol{\mathcal{X}}|\boldsymbol{s}^*, \lambda)$ (Gl. 2.10). Bei der Verwendung von überlagerten Mixturverteilungen $b_s(\boldsymbol{x}_t) = \sum_{k=1}^{K} c_{sk}\mathcal{N}(\boldsymbol{x}_t, \boldsymbol{\mu}_{sk}, \boldsymbol{\Sigma}_{sk}) = \sum_{k=1}^{K} c_{sk}\mathcal{N}_{sk}(\boldsymbol{x}_t) = \sum_{k=1}^{K} b_{sk}(\boldsymbol{x}_t)$ in einem Zustand muss obige Betrachtung jedoch erweitert werden. Durch Ausmultiplizieren aller möglichen Mixturfolgen $\boldsymbol{\Theta}_K \in \Omega_K$ (statt Zustandsfolgen) ergibt sich [Pla95]

$$\begin{aligned} p(\boldsymbol{\mathcal{X}}|\boldsymbol{s}^*, \lambda) &= \prod_{t=1}^{T} b_{s_t^*}(\boldsymbol{x}_t) = \prod_{t=1}^{T} \sum_{k=1}^{K} b_{s_t^* k}(\boldsymbol{x}_t) = \\ &= \sum_{k_1=1}^{K} \sum_{k_2=1}^{K} \cdots \sum_{k_T=1}^{K} \prod_{t=1}^{T} b_{s_t^* k_t}(\boldsymbol{x}_t) = \sum_{t=1}^{\Omega_K} \prod_{t=1}^{T} b_{s_t^* k_t}(\boldsymbol{x}_t) = \end{aligned}$$

$$= \sum^{\Omega_K} p(\boldsymbol{\mathcal{X}}, \boldsymbol{\Theta}_K | \boldsymbol{s}^*, \lambda) \tag{2.12}$$

Die Maximierung von $p(\boldsymbol{\mathcal{X}}|\boldsymbol{s}^*, \lambda)$ kann nach dem Maximum-Likelihood Prinzip unter Zuhilfenahme der Wachstumstransformation $Q(\lambda, \bar{\lambda})$ [Rab89] erreicht werden.

$$\begin{aligned} Q(\lambda, \bar{\lambda}) &= \sum^{\Omega_K} p(\boldsymbol{\mathcal{X}}, \boldsymbol{\Theta}_K | \boldsymbol{s}^*, \lambda) \log p(\boldsymbol{\mathcal{X}}, \boldsymbol{\Theta}_K | \boldsymbol{s}^*, \bar{\lambda}) = \\ &= \sum^{\Omega_K} p(\boldsymbol{\mathcal{X}}, \boldsymbol{\Theta}_K | \boldsymbol{s}^*, \lambda) \left[\sum_{t=1}^{T} \log \bar{\mathcal{N}}_{s_t^* k_t}(\boldsymbol{x}_t) + \sum_{t=1}^{T} \log \bar{c}_{s_t^* k_t}(\boldsymbol{x}_t) \right] \end{aligned} \tag{2.13}$$

Für die Optimierung der reinen Gaussverteilungsparameter genügt wiederum die Betrachtung der ersten Summationskomponente in Gl. 2.13.

$$\begin{aligned} Q(\lambda, \bar{\lambda}) &= \sum^{\Omega_K} p(\boldsymbol{\mathcal{X}}, \boldsymbol{\Theta}_K | \boldsymbol{s}^*, \lambda) \sum_{t=1}^{T} \log \bar{\mathcal{N}}_{s_t^* k_t}(\boldsymbol{x}_t) = \\ &= \sum_{i=1}^{N_S} \sum^{\Omega_K} p(\boldsymbol{\mathcal{X}}, \boldsymbol{\Theta}_K | \boldsymbol{s}^*, \lambda) \sum_{t=1}^{T} \log \bar{\mathcal{N}}_{s_t^* k_t}(\boldsymbol{x}_t) \delta(s_t^* - i) = \\ &= \sum_{i=1}^{N_S} \sum_{\kappa=1}^{K} \sum_{t=1}^{T} \log \bar{\mathcal{N}}_{i\kappa}(\boldsymbol{x}_t) \delta(s_t^* - i) \sum^{\Omega_K} p(\boldsymbol{\mathcal{X}}, \boldsymbol{\Theta}_K | \boldsymbol{s}^*, \lambda) \delta(k_t - \kappa) \end{aligned} \tag{2.14}$$

Definiert man $\gamma_{s_t^* k_t}$ als die Aposteriori-Wahrscheinlichkeit für Mixturkomponente k des Zustands s_t^* zum Zeitpunkt t

$$\gamma_{s_t^* k_t} = \frac{1}{p(\boldsymbol{\mathcal{X}}|\boldsymbol{s}^*, \lambda)} \sum^{\Omega_K} p(\boldsymbol{\mathcal{X}}, \boldsymbol{\Theta}_K, k_t | \boldsymbol{s}^*, \lambda) \tag{2.15}$$

so vereinfacht sich Gl. 2.14 zu

$$Q(\lambda, \bar{\lambda}) = p(\boldsymbol{\mathcal{X}}|\boldsymbol{s}^*, \lambda) \sum_{i=1}^{N_S} \sum_{\kappa=1}^{K} \sum_{t=1}^{T} \gamma_{i\kappa} \log \bar{\mathcal{N}}_{i\kappa}(\boldsymbol{x}_t) \delta(s_t^* - i) \tag{2.16}$$

damit wird die Optimierung äquivalent zum Fall einer unimodalen Verteilung [Jua90, Pla95]. Ausgehend von Gl. 2.16 werden in Abschnitt 2.4.2 und 2.5.3 die Nachschätzgleichungen der MLLR- sowie der MLED-Adaption abgeleitet.

2.3.2 Das "Maximum Likelihood"-Optimierungskriterium (ML)

Die direkte Maximum-Likelihood (ML) Schätzung der Verteilungsparameter stellt die bekannteste Form der Parameterschätzung dar. Charakteristisch für dieses Verfahren ist, dass in die Parameteroptimierung nur Daten des *eigenen* Klassenmodells eingehen - die Relation zu konkurrierenden Klassen wird nicht betrachtet. Dies steht im Gegensatz zu den diskriminativen Trainingsverfahren [ReW96], die auch rivalisierende Klassen in die Parameterschätzung einbeziehen. Bei der ML-Optimierung wird versucht die klassenspezifische Likelihood $p(\boldsymbol{\mathcal{X}}|\lambda)$ (Baum-Welch) bzw. $p(\boldsymbol{\mathcal{X}}, \boldsymbol{s}^*|\lambda)$ (Segmental K-Means) bzgl. der freien Parameter λ zu maximieren. Für das Segmental K-Means Prinzip ergibt sich aus Gl. 2.8

$$\bar{\lambda} = \arg\max_{\lambda} \left[\max_{\mathbf{s}} p(\boldsymbol{\mathcal{X}}, \boldsymbol{s}|\lambda) \right] = \arg\max_{\lambda} p(\boldsymbol{\mathcal{X}}, \boldsymbol{s}^*|\lambda) \tag{2.17}$$

Die Maximierung der Likelihood $p(\boldsymbol{\mathcal{X}}, \boldsymbol{s}^*|\lambda)$ kann iterativ unter Zuhilfenahme der Hilfsfunktion $Q(\lambda, \bar{\lambda})$ aus Gl. 2.13 erfolgen. Die Maximierung von Q ergibt sich bei $\frac{\partial}{\partial\lambda}Q = 0$. Ausführliche Herleitungen der Nachschätzgleichungen finden sich in [Pla95, Wol97]. Die Auswertung dieser Bedingung führt im Falle des verwendeten und implementierten Segmental K-Means Verfahrens zu folgenden Nachschätzgleichungen [Jua90, Pla95] für Mixturkoeffizienten, Mittelpunkte und Kovarianzmatrizen.

$$\bar{c}_{ik} = \frac{\sum_{t=1}^{T} \delta(s_t^* - i)\gamma(k|\boldsymbol{x}_t, i)}{\sum_{\kappa=1}^{K_i} \sum_{t=1}^{T} \delta(s_t^* - i)\gamma(\kappa|\boldsymbol{x}_t, i)} \tag{2.18}$$

$$\bar{\boldsymbol{\mu}}_{ik} = \frac{\sum_{t=1}^{T} \delta(s_t^* - i)\gamma(k|\boldsymbol{x}_t, i)\boldsymbol{x}_t}{\sum_{t=1}^{T} \delta(s_t^* - i)\gamma(k|\boldsymbol{x}_t, i)} \tag{2.19}$$

$$\bar{\boldsymbol{\Sigma}}_{ik} = \frac{\sum_{t=1}^{T} \delta(s_t^* - i)\gamma(k|\boldsymbol{x}_t, i)(\boldsymbol{x}_t - \bar{\boldsymbol{\mu}}_{ik})(\boldsymbol{x}_t - \bar{\boldsymbol{\mu}}_{ik})^T}{\sum_{t=1}^{T} \delta(s_t^* - i)\gamma(k|\boldsymbol{x}_t, i)} \tag{2.20}$$

Die vorangegangenen Nachschätzgleichungen wurden für den Fall einer einzelnen Trainingssequenz $\boldsymbol{\mathcal{X}}$ abgeleitet. Die Erweiterung der Gleichungen für R Sequenzen ist trivial und führt, neben der Summation über T, zu einer zusätzlichen Summation über alle R Sequenzen $\boldsymbol{\mathcal{X}}_r$. Aus Gründen der Übersichtlichkeit wird diese Schleife im folgenden jedoch vernachlässigt.

Beim Segmental K-Means Training gehen die Merkmalsvektoren $\boldsymbol{x}_t$ nur bei denjenigen Modellen, bzw. deren Zuständen in die Nachschätzgleichungen ein, denen sie laut Viterbisegmentierung hart zugeordnet wurden. Dies steht im Gegensatz zum Baum-Welch Training, bei dem die Merkmalsvektoren im Prinzip den Zuständen anhand einer Wahrscheinlichkeit weich zugeordnet werden. Der Term $\gamma(k|\boldsymbol{x}_t, i) = \gamma_{ik}(\boldsymbol{x}_t)$ ergibt sich aus Gl. 2.15 zu

$$\gamma(k|\boldsymbol{x}_t, i) = \frac{c_{ik}\mathcal{N}(\boldsymbol{x}_t|i, k)}{\sum_{k'=1}^{K_i} c_{ik'}\mathcal{N}(\boldsymbol{x}_t|i, k')} \tag{2.21}$$

Mittels dieser Gleichungen erfolgt iterativ die Annäherung an ein (lokales) Optimum der Likelihood $p(\boldsymbol{\mathcal{X}}, \boldsymbol{s}^*|\lambda)$

2.3.3 Das "Maximum Aposteriori"-Optimierungskriterium (MAP)

Beim Maximum-Likelihood Ansatz werden keinerlei Annahmen über die *Verteilung der Modellparameter* getroffen, d.h. sie werden als gleichverteilt angenommen. Dies steht im Gegensatz zu den Ansätzen der "Bayesian Estimation", unter deren Kategorie auch das sogenannte

Maximum Aposteriori-Training [Gau92, LeC93, Pfa98b, Pfa00b] fällt. Als zu optimierendes Zielkriterium wird hier die Aposteriori-Wahrscheinlichkeit $p(\lambda|\boldsymbol{\mathcal{X}})$ der Äußerung $\boldsymbol{\mathcal{X}}$ verwendet. Unter Zuhilfenahme der Gleichung von Bayes lässt sich die Zielfunktion umformen zu:

$$\bar{\lambda} = \arg\max_{\lambda} p(\lambda|\boldsymbol{\mathcal{X}}) = \arg\max_{\lambda} p(\lambda)p(\boldsymbol{\mathcal{X}}|\lambda) \tag{2.22}$$

In die Optimierung geht hier die Apriori-Wahrscheinlichkeit der Parameter λ ein. Analog zur Herleitung der ML-Gleichungen kann auch hier die Maximierung über das Segmental K-Means Prinzip erfolgen [Gau92]. Eine ausführliche Herleitung und Diskussion des MAP-Ansatzes findet sich in [Pfa00b]. Die MAP-Optimierung führt zu folgenden Nachschätzgleichungen:

$$\bar{c}_{ik} = \frac{(\nu_{ik}-1) + \sum_{t=1}^{T} \delta(s_t^*-i)\gamma(k|\boldsymbol{x}_t,i)}{\sum_{\kappa=1}^{K_i}(\nu_{ik}-1) + \sum_{\kappa=1}^{K_i}\sum_{t=1}^{T} \delta(s_t^*-i)\gamma(\kappa|\boldsymbol{x}_t,i)} \tag{2.23}$$

$$\bar{\boldsymbol{\mu}}_{ik} = \frac{\tau_{ik}\boldsymbol{m}_{ik} + \sum_{t=1}^{T} \delta(s_t^*-i)\gamma(k|\boldsymbol{x}_t,i)\boldsymbol{x}_t}{\tau_{ik} + \sum_{t=1}^{T} \delta(s_t^*-i)\gamma(k|\boldsymbol{x}_t,i)} \tag{2.24}$$

$$\begin{aligned}\bar{\boldsymbol{\Sigma}}_{ik} = {} & \frac{\sum_{t=1}^{T} \delta(s_t^*-i)\gamma(k|\boldsymbol{x}_t,i)(\boldsymbol{x}_t-\bar{\boldsymbol{\mu}}_{ik})(\boldsymbol{x}_t-\bar{\boldsymbol{\mu}}_{ik})^T}{(\alpha_{ik}-p) + \sum_{t=1}^{T} \delta(s_t^*-i)\gamma(k|\boldsymbol{x}_t,i)} + \\ & + \frac{\boldsymbol{u}_{ik} + \tau_{ik}(\boldsymbol{m}_{ik}-\bar{\boldsymbol{\mu}}_{ik})(\boldsymbol{m}_{ik}-\bar{\boldsymbol{\mu}}_{ik})^T}{(\alpha_{ik}-p) + \sum_{t=1}^{T} \delta(s_t^*-i)\gamma(k|\boldsymbol{x}_t,i)}\end{aligned} \tag{2.25}$$

Bei der Implementierung der MAP-Nachschätzgleichungen wird häufig der sog. Empirisch-Bayes'sche Ansatz vorgezogen. Hierbei werden die Hyperparameter ν_{ik},α_{ik},$\boldsymbol{u}_{ik}$ und $\boldsymbol{m}_{ik}$ ebenso aus der vorliegenden Datenverteilung geschätzt. Der Vergleich der ML-Nachschätzgleichungen (Gl. 2.18-2.20) mit ihren MAP-Äquivalenten (Gl. 2.23-2.25) zeigt gewisse Ähnlichkeiten auf. So können die MAP-Gleichungen als eine gewichtete Mittelung zwischen einem Apriori-Anteil und der ML-Schätzung aufgefasst werden. Bei sehr wenig Trainings- bzw. Adaptionsdaten überwiegt der Apriori-Anteil, und es kommt nur zu geringfügigen Änderungen gegenüber dem Apriori-Modell. Stehen sehr viele Daten zur Verfügung, dann konvergieren die Parameter gegen die ML-Schätzung. Gerade diese Eigenschaft lässt den MAP-Ansatz als Adaptionsalgorithmus geeignet erscheinen. Unter Verwendung generischer, z.B. sprecherunabhängiger HMM-Modelle als Apriori-Modell kann mit vergleichsweise wenig Daten eine Anpassung an veränderte Rahmenbedingungen, wie z.B. den aktuellen Sprecher, erfolgen. Vorteilhafterweise werden die in den Adaptionsdaten ungesehenen Modelle nicht nachteilig verändert. Allerdings bleibt zu beachten, dass der Algorithmus zwar einerseits sehr robust ist, andererseits aber auch eine sehr langsame Konvergenz aufweist. Gerade zur schnel-

len Sprecheradaption, bei der anwendungsbedingt nur wenige Sekunden Adaptionsdaten zur Verfügung stehen, ist die Anpassung u.U. nicht ausreichend genug.

2.4 Das "Maximum Likelihood Linear Regression"-Verfahren (MLLR)

2.4.1 Einführung

Ein zentrales Problem bei der Erzeugung von sprecherangepassten Modellen ist in der Regel die Verfügbarkeit von ausreichend Trainingsdaten. Bei Diktiersystemen beispielsweise kann es einem Anwender aus Akzeptanzgründen nicht zugemutet werden, lange damit zuzubringen definierte Sprachdaten einzugeben, um das System anzupassen. Bei Telefonauskunftssystemen andererseits ist die Gesprächsdauer meist zu kurz, um genügend Daten zu sammeln, die für eine robuste ML-Neuschätzung der Systemparameter nötig wären. Im Extremfall muss hier aus wenigen Sekunden Sprache (100-1000 Merkmalsvektoren) auf mehrere Hunderttausend Parameter (z.B. Tab. 5.1, 8500 Normalverteilungen mit jeweils 42 Dimensionen) geschlossen werden. Bei HMM-Modellen ist der ML-Ansatz in Kombination mit einem Segmental K-Means Ansatz [Jua90] hierzu ungeeignet, da er ausschließlich die Adaptionsdaten zur Parameterschätzung (s. Gl. 2.18-2.20) berücksichtigt. Als Beispiel: bei einer angenommenen Gleichverteilung würden bei 44 HMMs mit je 3 Zuständen auf jeden Zustand eines HMMs demnach $\frac{1000}{44*3} \approx 7$ Vektoren entfallen. Bei ca. 64 Normalverteilungen je Zustand ist eine robuste Parameterschätzung mit diesem Verfahren nicht möglich.

Eine Adaption mittels MAP bietet auch nur einen bedingten Ausweg, da die Berücksichtigung der Verteilung der Systemparameter zwar zu robusten Nachschätzgleichungen (s. Gl. 2.23-2.25) führt, diese jedoch eine sehr langsame Konvergenz aufweisen.

Kern der meisten Adaptionsverfahren ist die Einbeziehung von Zusatzinformation, um den Freiheitsgrad der zu schätzenden Parameter einzuschränken. Typisch hierfür ist das Konzept einer gemeinsamen Translation, wie es beispielsweise von Furui [Fur89] vorgeschlagen wurde. Er benutzte Datenzentroide, um Verschiebungsvektoren für die Parameter eines Codebuchs zu berechnen. Die Annahme einer für die Unbekannten gleichen Veränderung ist ein effektiver Weg, um die Zahl der freien Parameter zu reduzieren [Tak95, Fab97]. Beim "Maximum Likelihood Linear Regression"-Adaptionsverfahren (MLLR) wird dieser Ansatz durch eine gemeinsame lineare Transformation $\boldsymbol{A}_r$ mit $r = 1..K_R$ der Mittelpunktsvektoren umgesetzt [Leg95].

$$\hat{\boldsymbol{\mu}}_{ik} = \boldsymbol{A}_r \boldsymbol{\mu}_{ik} \tag{2.26}$$

Diese Transformation lässt sich durch die Einführung einer optionalen Verschiebung zu einer affinen Transformation erweitern:

$$\hat{\boldsymbol{\mu}}_i = \boldsymbol{A}_r \boldsymbol{\mu}_{ik} + \boldsymbol{b}_r = \boldsymbol{W}_r \tilde{\boldsymbol{\mu}}_{ik} \tag{2.27}$$

mit $\tilde{\boldsymbol{\mu}}_{ik}^T = [1\ \boldsymbol{\mu}_{ik}^T]$ und $\boldsymbol{W}_r = [\boldsymbol{b}_r\ \boldsymbol{A}_r]$. Der Vorteil dieser Art der Adaption liegt darin, dass durch gemeinsame Regressionen auch Modellparameter für die keine Adaptionsdaten vorlie-

gen, einer Anpassung unterzogen werden können. Sie unterliegen dann derselben Regression, wie ähnliche Modelle, für die jedoch Daten gesehen wurden. Ein ähnlicher transformationsorientierter Ansatz findet sich bei Digalakis et al. [Dig95a, Dig95b].

Im Vergleich zur Gesamtzahl der Systemparameter ergibt sich durch die gemeinsame Regression eine deutlich reduzierte Anzahl an zu schätzenden Parametern. So sind bei K_R verwendeten Transformationsmatrizen ("Regressionsklassen") insgesamt $K_R * (N_D \times N_D + N_D)$ freie Parameter zu bestimmen. Dies sind i.d.R. erheblich weniger als die Zahl der Modellparameter. Bei der Annahme von nur einer globalen Regression, d.h. $K_R = 1$, werden alle Parameter derselben identischen Transformation unterzogen:

$$\boldsymbol{W}_r = \boldsymbol{W} \tag{2.28}$$

für $i = 1 \ldots N_S$. Damit reduziert sich die Zahl der unbekannten Parameter auf $N_D \times (N_D + 1)$. Bei einer Vektordimension von 42, wie es beispielsweise bei der Verwendung der MFCC42-Vorverarbeitung der Fall ist, ergeben sich also $42 * 43 = 1806$ zu bestimmende Koeffizienten. Bei einer angenommenen Erkennergröße von 8500 Verteilungen, d.h. ca. 357000 Mittelpunktsparametern, entspricht dies einer Reduktion von $1806 : 357000 \approx 1 : 200$. Diese vergleichsweise wenigen Parameter lassen sich auch bei kurzen Adaptionssequenzen sehr robust schätzen.

2.4.2 ML-Schätzung der Transformationsmatrix

Die Koeffizienten der Transformationsmatrizen W_r lassen sich über eine ML-Schätzung [Leg95] ableiten. Die Maximierung der Likelihood $p(\boldsymbol{\mathcal{X}}, \boldsymbol{s}^*|\lambda)$ kann unter Zuhilfenahme der Hilfsfunktion $Q(\lambda, \bar{\lambda})$ aus Gl. 2.13 erfolgen. Für den Fall Gauss'scher Mixturverteilungen in den einzelnen HMM-Zuständen, konnte die Hilfsfunktion, wie in Abschnitt 2.3.1 gezeigt, schließlich auf die Form in Gl. 2.16 gebracht werden. Für die Maximierung muss gelten: $\frac{\partial}{\partial\lambda}Q(\lambda, \bar{\lambda})=0$. Als zu optimierende Parameter λ sind hier die Koeffizienten der Transformationsmatrizen $\boldsymbol{W}_r$ zu bestimmen. Ausgehend von Gl. 2.16 ergibt sich:

$$\begin{aligned}
\frac{\partial Q}{\partial \boldsymbol{W}_r} &= p(\boldsymbol{\mathcal{X}}|\boldsymbol{s}^*, \lambda) \sum_{\kappa=1}^{K} \sum_{i \in \Omega_r} \sum_{t=1}^{T} \gamma_{i\kappa} \delta(s_t^* - i) \frac{\partial}{\partial \boldsymbol{W}_r} (\log \bar{\mathcal{N}}_{i\kappa}(\boldsymbol{x}_t)) = \\
&= p(\boldsymbol{\mathcal{X}}|\boldsymbol{s}^*, \lambda) \sum_{\kappa=1}^{K} \sum_{i \in \Omega_r} \sum_{t=1}^{T} \gamma_{i\kappa} \delta(s_t^* - i) \frac{1}{\bar{\mathcal{N}}_{i\kappa}(\boldsymbol{x}_t)} \bar{\mathcal{N}}_{i\kappa}(\boldsymbol{x}_t) \\
&\quad * \frac{\partial}{\partial \boldsymbol{W}_r} \left(-\frac{1}{2} (\boldsymbol{x}_t - \boldsymbol{W}_r \tilde{\boldsymbol{\mu}}_{i\kappa})^T \boldsymbol{\Sigma}_{i\kappa}^{-1} (\boldsymbol{x}_t - \boldsymbol{W}_r \tilde{\boldsymbol{\mu}}_{i\kappa}) \right) = \\
&= p(\boldsymbol{\mathcal{X}}|\boldsymbol{s}^*, \lambda) \sum_{\kappa=1}^{K} \sum_{i \in \Omega_r} \sum_{t=1}^{T} \gamma_{i\kappa} \delta(s_t^* - i) \boldsymbol{\Sigma}_{i\kappa}^{-1} (\boldsymbol{x}_t - \boldsymbol{W}_r \tilde{\boldsymbol{\mu}}_{i\kappa}) \tilde{\boldsymbol{\mu}}_{i\kappa}^T = 0
\end{aligned} \tag{2.29}$$

Die Summation erfolgt in Gl. 2.29 jedoch nur noch über die Zustände $i \in \Omega_r$, die durch die gemeinsame Regression $\boldsymbol{W}_r$ erfasst werden. Nur im Falle einer globalen Regression $\boldsymbol{W}_r = \boldsymbol{W}$ würden alle N_S Zustände eingeschlossen. Sortiert man in Gl. 2.29 die von $\boldsymbol{W}_r$ abhängigen Anteile auf die rechte Seite, die unabhängigen Anteile nach links, so ergibt sich

$$\sum_{\kappa=1}^{K}\sum_{i\in\Omega_r}\sum_{t=1}^{T}\gamma_{i\kappa}\delta(s_t^*-i)\boldsymbol{\Sigma}_{i\kappa}^{-1}\boldsymbol{x}_t\tilde{\boldsymbol{\mu}}_{i\kappa}^T=\sum_{\kappa=1}^{K}\sum_{i\in\Omega_r}\sum_{t=1}^{T}\gamma_{i\kappa}\delta(s_t^*-i)\boldsymbol{\Sigma}_{i\kappa}^{-1}\boldsymbol{W}_r\tilde{\boldsymbol{\mu}}_{i\kappa}\tilde{\boldsymbol{\mu}}_{i\kappa}^T \tag{2.30}$$

Nach [Leg95] lassen sich definieren:

$$\boldsymbol{V}^{i\kappa} = \sum_{t=1}^{T}\gamma_{i\kappa}\delta(s_t^*-i)\boldsymbol{\Sigma}_{i\kappa}^{-1} \tag{2.31}$$

$$\boldsymbol{D}^{i\kappa} = \tilde{\boldsymbol{\mu}}_{i\kappa}\tilde{\boldsymbol{\mu}}_{i\kappa}^T \tag{2.32}$$

Damit wird Gl. 2.30 zu

$$\sum_{\kappa=1}^{K}\sum_{i\in\Omega_r}\sum_{t=1}^{T}\gamma_{i\kappa}\delta(s_t^*-i)\boldsymbol{\Sigma}_{i\kappa}^{-1}\boldsymbol{x}_t\tilde{\boldsymbol{\mu}}_{i\kappa}^T=\sum_{\kappa=1}^{K}\sum_{i\in\Omega_r}\boldsymbol{V}^{i\kappa}\boldsymbol{W}_r\boldsymbol{D}^{i\kappa} \tag{2.33}$$

Schlüsselt man die rechte Seite der Gleichung nach den einzelnen Matrixelementen auf, so führt dies zu

$$y_{mn}=\sum_{p=1}^{N_D}\sum_{q=1}^{N_D+1}w_{pq}^r\left(\sum_{\kappa=1}^{K}\sum_{i\in\Omega_r}v_{mp}^{i\kappa}d_{qn}^{i\kappa}\right) \tag{2.34}$$

Aufgrund der Symmetrie von $\boldsymbol{D}^{i\kappa}$, sowie der vorliegenden Diagonalität der Kovarianzmatrizen

$$g_{qn}^{(m)}=\sum_{\kappa=1}^{K}\sum_{i\in\Omega_r}v_{mp}^{i\kappa}d_{qn}^{i\kappa}=\begin{cases}\sum_{\kappa=1}^{K}\sum_{i\in\Omega_r}v_{mm}^{i\kappa}d_{qn}^{i\kappa} & \text{falls} \quad m=p\\ 0 & \text{sonst}\end{cases} \tag{2.35}$$

reduziert sich die Summation über p in Gl. 2.34 auf den Beitrag bei $p=m$. Die linke Seite von Gl. 2.33 lässt sich unabhängig von $\boldsymbol{W}_r$ berechnen. Fasst man deren Ergebnis in der resultierenden Matrix $\boldsymbol{Z}$ zusammen, so führt die Betrachtung der Einzelelemente z_{mn} zu

$$z_{mn}=y_{mn}=\sum_{q=1}^{N_D+1}w_{mq}^r g_{qn}^{(m)}=(\boldsymbol{w}_m^r)^T\boldsymbol{g}_n^{(m)} \tag{2.36}$$

$$\boldsymbol{z}_m^T=(\boldsymbol{w}_m^r)^T\boldsymbol{G}^{(m)} \quad \text{mit} \quad \boldsymbol{G}^{(m)}=\left[\boldsymbol{g}_1^{(m)}\ \boldsymbol{g}_2^{(m)}\ \ldots\ \boldsymbol{g}_{N_D+1}^{(m)}\right] \tag{2.37}$$

Die Gleichungen lassen sich für $m=1..N_D$ zeilenweise lösen. Aus den Einzellösungen

$$(\boldsymbol{w}_m^r)^T=(\boldsymbol{G}^{(m)})^{-1}(\boldsymbol{z}_m)^T \tag{2.38}$$

kann die Gesamtmatrix $\boldsymbol{W}_r$ rekonstruiert werden.

2.5 "Eigenvoices" und das "Maximum Likelihood Eigenspace Decomposition"-Adaptionsverfahren (MLED)

2.5.1 Eigenvoices

Zur Reduktion der zu schätzenden, freien Parameter wird beim MLLR-Trainingsverfahren eine gemeinsame, affine Transformation der Parameter *angenommen*. Im Gegensatz einer reinen Annahme wird beim sogenannten Eigenvoice-Ansatz [Kuh98, Kuh99, Kuh00, Ngu99, Fal01a], bzw. dem darauf aufbauenden Adaptionsverfahren, explizites Wissen über die sprecherspezifische Verteilung der Modellparameter zur Parameteroptimierung ausgewertet. Ausgangspunkt ist hier die Analyse der Veränderung der Modellparameter über verschiedene Sprecher hinweg. Der Ursprung dieses Ansatzes ist in der Bilderkennung - speziell Gesichtserkennung - begründet [Tur91a, Tur91b, Pen94]. Hier hat sich gezeigt, dass unterschiedliche Gesichter nicht überall gleich stark in der Bildinformation voneinander abweichen, sondern primär in gewissen Regionen und darüber hinaus in bestimmter Art und Weise. Fasst man jedes Bild als *einen* Punkt im $N*M$-dimensionalen Raum ($N \times M$-Bildpunkte (2-dim) $\rightarrow N*M \times 1$ Vektor (1-dim)) auf, so ergeben sich vorzugsweise Raumrichtungen in denen sich die Gesichter unterscheiden. Auf Basis dieser Streurichtungen lässt sich ein beliebiges Gesicht aus einem "mittlerem" Gesicht und einer gewichteten Überlagerung dieser Raumrichtungen konstruieren. Die Streurichtungen werden als Eigengesichter (engl.: 'Eigenfaces') bezeichnet. Die Eigengesichter stellen dabei die Richtungen der Hauptabweichungen, d.h. der Varianz dar. Übertragen auf die Spracherkennung bedeutet dieser Ansatz, dass sich ein beliebiges Sprechermodell aus einem "mittleren Sprechermodell" und der gewichteten Überlagerung in Richtung der Hauptabweichungen zwischen den Sprechermodellen darstellen lässt. In Analogie zur Gesichtserkennung werden die Hauptstreurichtungen der Sprechermodellparameter als *Eigenvoices* bezeichnet.

2.5.2 Generierung des Eigenraums

Die Modellparameter $\boldsymbol{\lambda}^m$ (als Vektor) für einen Sprecher m ergeben sich aus den mittleren Parametern $\boldsymbol{m}_0$ und der gewichteten Überlagerung der Basisrichtungen $\boldsymbol{e}_k$.

$$\boldsymbol{\lambda}^m = \boldsymbol{m}_0 + \sum_{k=1}^{K_{EV}} w_k^m \boldsymbol{e}_k \tag{2.39}$$

Die Basisrichtungen=Eigenvoices $\boldsymbol{e}_k$ müssen aus der Analyse sprecherabhängiger Modelle gewonnen werden. Wenn für jeden Sprecher aus einer Trainingsdatenbank ein eigenes Modell trainiert wurde, kann aus den mittleren Abweichungen der Sprechermodelle gegenüber dem Normsprechermodell Rückschlüsse auf die Hauptabweichungsrichtungen gezogen werden. Der Parameter-"Supervektor" $\boldsymbol{\lambda}^m$ hat die Dimension $N_D^{EV} \times 1$, wobei N_D^{EV} der Summe aller freien Parameter entspricht, bzw. entsprechen kann, falls alle unterschiedlichen Parametertypen (Mittelpunkte,Varianzen,Mixturkoeffizienten) berücksichtigt werden:

$$N_D^{EV} = N_{Bf} * (N_D * 2 + 1) = (N_D * 2 + 1) \sum_{i=1}^{N_S} N_{Bf}^i \tag{2.40}$$

Es wurden hierbei eine zustandsweise Gauss'sche Mixturverteilungen mit insgesamt N_{Bf} Mixturen und die Verwendung diagonaler Kovarianzen angenommen. Im Gegensatz zur Bil-

derkennung ergibt sich hier jedoch ein zusätzliches Problem bei der "Vektorisierung" der Modellparameter $\{\lambda\}_m \rightarrow \boldsymbol{\lambda}^m$. Bei Bildern ist, nach Ausrichtung und Skalierung des Gesichts, die Abfolge der Bildpunkte in Zeilen und Spalten fest vorgegeben. Die Vektorisierung kann durch einfaches Transformieren und Umsortieren der Zeilen- in Spaltenvektoren erfolgen. Unter Beibehaltung der Transformationsreihenfolge bleibt *die Bedeutung* der transformierten Einträge über verschiedene Ausgangsbilder hinweg erhalten. Die Bildpunkte der Nase beispielsweise tauchen im Supervektor dementsprechend für verschiedene Bilder jeweils an annähernd der gleichen Stelle auf. Bei der Übertragung auf Sprechermodelle ist die bedeutungsmäßige Zuordnung der Parameter zwischen den Sprechern u.U. nicht gewährleistet, da diese mit unterschiedlichen Daten geschätzt werden und somit unterschiedliche Lagen im Merkmalsraum einnehmen. Ein Kernaspekt bei der Konstruktion von Eigenvoices liegt daher im geeigneten Aufbau der Supervektoren $\boldsymbol{\lambda}^m$, sowie der zugrundeliegenden Sprechermodelle, mit dem Ziel die Bedeutung die Koeffizienten zwischen Sprechern konsistent zuordnen zu können.

2.5.2.1 Supervektoren

Zur Analyse der Parameterabweichungen der Einzelsprecher ist es günstig, die Organisationsstruktur der Parameter in ein Vektorformat umzuordnen. Als Kenngrößen sind die N_{Bf} Normalverteilungen normalerweise auf die Einzelzustände der Hidden-Markov-Modelle verteilt und dort "parallel" jeweils mit Dimension N_D "angeordnet". Sie werden daher in einer vorgegebenen Reihenfolge aus dieser flachen Struktur in einem einzelnen, gemeinsamen Supervektor zusammengefasst (s. Abb. 2.1). Die folgende Betrachtung konzentriert sich auf die Mittelpunktskoeffizienten $\boldsymbol{\mu}$ der Gaussverteilungen.

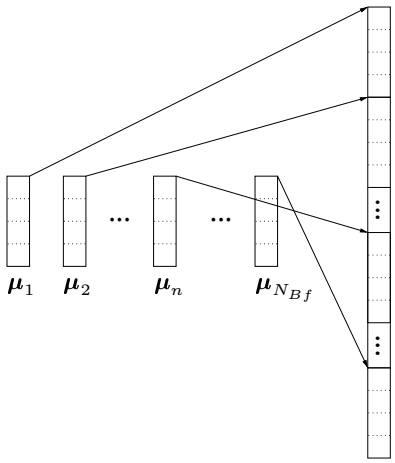

Abb. 2.1: Ausrichtung von N_{Bf} Mittelpunktsvektoren der Dimension N_D in einem gemeinsamen Supervektor $\boldsymbol{z}$ der Dimension $N_D^{EV} = N_{Bf} * N_D$.

Wie Abb. 2.1 zeigt, wird ein Supervektor $\boldsymbol{z}$ sequentiell aus den Einzelmittelpunktsvektoren zusammengesetzt. Bei N_{Bf} Mittelpunkten der Dimension N_D hat der Supervektor - unter der Voraussetzung, dass nur die Mittelpunkte für die Bestimmung der Eigenstimmen

herangezogen werden - daher die Dimension $N_{Bf} * N_D$. Prinzipiell könnten jedoch alle Parameter eines Modells, d.h. Mittelpunkte, (Ko-)Varianzen sowie Mixturkoeffizienten in die Berechnung einbezogen werden. Die (Mittelpunkts-)Parameter werden der Reihe nach im Supervektor 'aufgefädelt'. Die Zählung beginnt bei den Verteilungen in Zustand 1 des HMM-Modells 1, setzt sich bei Zustand 2 des HMMs 1 fort und endet bei im letzten Zustand des HMMs N_{Ph}. Kernproblem hierbei ist, dass die Reihenfolge - bzw. anders ausgedrückt - die 'Bedeutung' der einzelnen Mittelpunkte, zwischen den Sprechern beibehalten werden muss. Gesucht ist hier die Abhängigkeit einer Merkmalskomponente vom Sprecher, bzw. die Abhängigkeit (Korrelation) mit anderen Komponenten des Merkmalsvektors. Bei der beschriebenen Art die Mittelpunkte anzuordnen, betrifft dies insbesondere die Mittelpunkte innerhalb eines Zustands. Beispielhaft sei dies für 2 Modellmittelsvektoren $\boldsymbol{\mu}_1^A$,$\boldsymbol{\mu}_2^A$ respektive $\boldsymbol{\mu}_1^B$,$\boldsymbol{\mu}_2^B$ zweier Sprecher A und B gezeigt, wobei der Index die gleiche "Bedeutung" andeuten soll. Ist keine Beziehung der Mittelpunkte zueinander bekannt, kann keine vernünftige Reihenfolge der Modellparameter angebeben werden. So ließen sich selbst bei obigem, einfachen Beispiel bereits 2 mögliche unterschiedliche Kombinationen angeben:

$$\begin{pmatrix}\boldsymbol{\mu}_1^B\\ \boldsymbol{\mu}_2^B\end{pmatrix} \leftrightarrow \begin{pmatrix}\boldsymbol{\mu}_1^A\\ \boldsymbol{\mu}_2^A\end{pmatrix} \quad bzw. \quad \begin{pmatrix}\boldsymbol{\mu}_1^B\\ \boldsymbol{\mu}_2^B\end{pmatrix} \leftrightarrow \begin{pmatrix}\boldsymbol{\mu}_2^A\\ \boldsymbol{\mu}_1^A\end{pmatrix}$$

Wobei hier nur die erste Kombination sinnvolle Aussagen erlaubt. Die Zahl der Kombinationen steigt mit der Anzahl der Mittelpunkte. Gefordert ist nun ein Ansatz der dafür sorgt, dass in den resultierenden Supervektoren jeweils ähnliche Mittelpunkte an der gleichen Position im Vektor zu liegen kommen, d.h. annähernd die Zuordnung $\boldsymbol{\mu}_1^A$-$\boldsymbol{\mu}_1^B$-$\boldsymbol{\mu}_1^C$-... erreicht wird. Im Prinzip ließe sich die (gleiche) Bedeutung von Basisfunktion verschiedener Sprecher über Ähnlichkeits- oder Abstandsmaße bestimmen. Bei sehr vielen Sprechern wird die Bestimmung rasch sehr komplex und v.a. nicht mehr eineindeutig. Aus diesem Grund hat sich die Ableitung der sprecherspezifischen Modelle aus einem generischen, sprecherunabhängigen Modell durchgesetzt (s. Abb. 2.2).

In der Abbildung ist dies angedeutet durch die entsprechenden Mittelpunkte $\boldsymbol{\mu}_1^G$ und $\boldsymbol{\mu}_2^G$ des sprecherunabhängigen Modells G. Aus diesem generischen Modell können, durch Nachtraining mit den sprecherspezifischen Daten, sprecherabhängige Modellsätze abgeleitet werden. Als Trainingsverfahren kann hier das Maximum Aposteriori-Training [Gau92] Anwendung finden. Die sich ergebenden Nachschätzgleichungen (Gl. 2.24) dieses Trainingsverfahrens beschreiben im Prinzip ein gewichtetes Mittel zwischen dem generischen Ausgangsmittelpunkt und der ML-Schätzung der neuen, sprecherspezifischen Daten. Bedingt durch die Tatsache, dass jede Verteilung, bzw. deren Mittelpunkt, eindeutig aus dem jeweiligen Pendant des generischen Modells hervorgegangen ist, kann die nötige Referenz zwischen Prototyp und gewählter Position im Supervektor zwischen den Sprechern aufrecht erhalten werden. Allerdings ist auch hier zu beachten, dass zu viele Trainingsiterationen zu einer Durchmischung der Prototypen führen können und damit die erhaltene Referenz die gewünschte Bedeutung verliert.

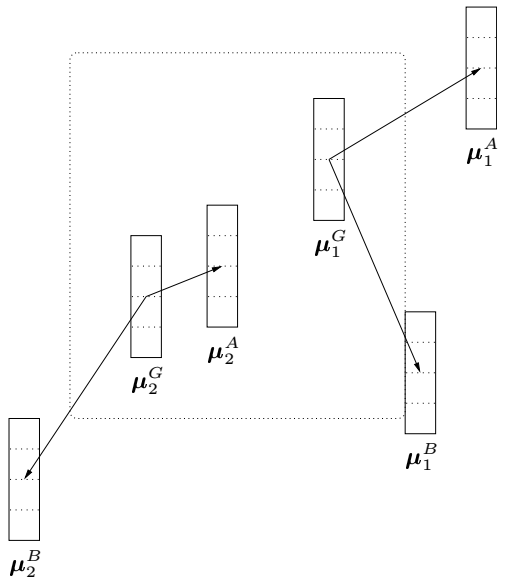

Abb. 2.2: Ableitung sprecherspezifischer Prototypen aus einem generischen Modell, schematisch gezeigt für 2 Sprecher A und B.

2.5.2.2 Eigenraumzerlegung mittels PCA

Ziel des Eigenstimmen-Ansatzes ist es, diejenigen Richtungen im Merkmalsraum (hier: Raum bzgl. der Modellparameter) zu finden, in denen die größten Abweichungen zwischen den Sprechern auftreten. Der Hintergedanke ist, das Apriori-Wissen, innerhalb welcher Grenzen sich die Modellparameter der Sprecher bewegen, bei der Generierung von neuen, sprecherspezifischen Modellen, als Einschränkung für den Freiheitsgrad der zu bestimmenden Parameter, verwenden zu können. Ausgangspunkt dafür ist das Finden derjenigen Raumrichtungen, in denen die größten Streuungen innerhalb der Sprecherpopulation auftreten.

Durch die Anwendung einer Hauptachsentransformation (PCA, engl. "Principle Components Analysis") [Rus94] kann für eine gegebene Mustervektormenge eine Vektorbasis gefunden werden, in der die Dimensionen bezüglich der Vektormenge dekorreliert sind. Die PCA liefert einen neuen Merkmalsraum, dessen Achsen entlang der Hauptstreurichtungen der Verteilung ausgerichtet sind. Abb. 2.3 verdeutlicht diesen Zusammenhang für den 2-dimensionalen Fall.

Für den in Abb. 2.3 dargestellten Fall ist $N_D^{org} = N_D^{PCA}$, d.h. die Dimension des gedrehten Raums ist gleich der Dimension des Originalraums. Prinzipiell gilt hierbei immer:

$$N_D^{PCA} \leq N_D^{org} \tag{2.41}$$

Durch Weglassen von Achsen mit geringer Varianz kann die Dimension des entstehenden Raums reduziert werden. Nach oben hin ist sie jedoch durch die Dimension des Originalraums begrenzt. Diese "Einschränkung" gewinnt praktisch gesehen dann an Bedeutung, wenn einerseits die Modelle sehr klein (wenige Verteilungen) sind und andererseits die Dimension der Verteilungen niedrig (z.B. keine Delta-Koeffizienten) ist. Dies kann beispielsweise bei

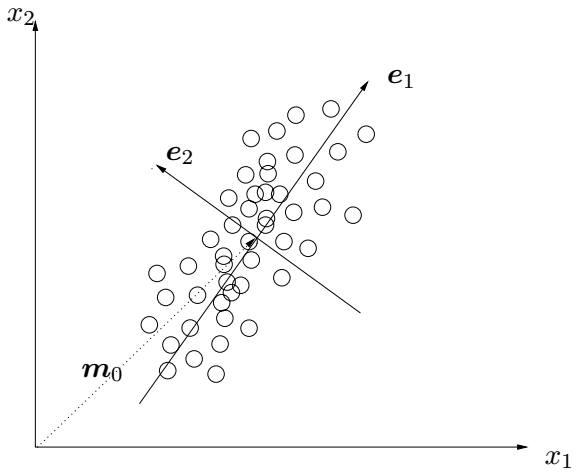

Abb. 2.3: Hauptachsen einer 2-dimensionalen Musterverteilung.

GMM-Modellen zur Sprechererkennung [Fal01a] der Fall sein. Die Kovarianzmatrix $\boldsymbol{C}_Z$ der Verteilung ist definiert als das Moment 2. Ordnung:

$$\boldsymbol{C}_Z = \frac{1}{K_S}\sum_{j=1}^{K_S}(\boldsymbol{z}_j - \boldsymbol{m}_0)(\boldsymbol{z}_j - \boldsymbol{m}_0)^T = \frac{1}{K_S}\sum_{j=1}^{K_S}\boldsymbol{z}_j\boldsymbol{z}_j^T - \boldsymbol{m}_0\boldsymbol{m}_0^T \tag{2.42}$$

wobei der Mittelwert $\boldsymbol{m}_0$ aller Sprecher K_S sich zu

$$\boldsymbol{m}_0 = \frac{1}{K_S}\sum_{j=1}^{K_S}\boldsymbol{z}_j \tag{2.43}$$

ergibt. Da alle Normalverteilungsparameter der Modelle eines Sprechers j zu einem Vektor zusammengefasst werden, wird jeder Sprecher durch genau *einen* Supervektor $\boldsymbol{z}_j$ repräsentiert. Zur Unterscheidung von den vergleichsweise niedrigdimensionalen HMM-Mittelpunkten ($\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$) seien die Beschreibungsparameter der Supervektorverteilung mit $\boldsymbol{m}_0$ und $\boldsymbol{C}_Z$ bezeichnet. Die Berechnung der Hauptachsen [Rus94] beruht im Prinzip auf einer Eigenwertzerlegung der Streumatrix $\boldsymbol{C}_Z$.

$$\boldsymbol{\Lambda}_Z = \boldsymbol{E}_{EV}^T\boldsymbol{C}_Z\boldsymbol{E}_{EV} = diag\{\lambda_{Zi}\} = diag\{\sigma_{Zi}^2\} \tag{2.44}$$

Die Transformationsmatrix $\boldsymbol{E}_{EV}$ beschreibt das neue, gedrehte Koordinatensystem. Zu beachten ist hierbei jedoch die mögliche Dimension des neuen Systems: bei den Modellen auf die dieses Verfahren normalerweise Anwendung finden kann - z.B. Hidden-Markov-Modelle - gilt i.d.R. $N_D^{EV} = Dim(\boldsymbol{z}) \gg K_S$, d.h. die Anzahl der Parameter $Dim(\boldsymbol{z})$ ist sehr viel größer als die Anzahl der verfügbaren Sprecher. Damit hat die Matrix $\boldsymbol{C}_Z$ den Rang

$$K_{EV}^{max} = Rg(\boldsymbol{C}_Z) = K_S - 1 \tag{2.45}$$

Im gedrehten Raum existieren also nur $K_S - 1$ linear unabhängige Basisvektoren (Hauptachsen). Bei der numerischen Berechnung ist ein weiterer Aspekt von essentieller Bedeutung. Die für eine Eigenwertzerlegung anwendbaren Verfahren (z.B. QR-Zerlegung) sind in der

Komplexität von mehr als quadratischer Ordnung. Dies lässt die Matrixgröße zu einem zentralen Problem werden. Als Beispiel: die Eigenwertzerlegung einer Matrix der Dimension 7000×7000 mittels des verwendeten Programmpakets, das auf einer Householder-Zerlegung basiert, dauert auf einem Pentium II 400 MHz System mehrere Stunden. Die volle Matrix $\boldsymbol{C}_Z$ eines Erkennungssystems hat jedoch die Dimension $N_D^{EV} \times N_D^{EV}$, wobei N_D^{EV} bereits bei einem relativ kleinen System mit beispielsweise 8500 Normalverteilungen (vgl. Tab. 5.1) $N_D^{EV} = 8500 * 42 \approx 3.6 * 10^5$ erreicht (bei MFCC42). Allein das Speichern dieser Matrix mit Float-Variablen würde $(N_D^{EV})^2 * 4Byte \approx 5.2 * 10^{11} Byte \approx 480Gb$ (!) Hauptspeicher erfordern. Um die Rechenzeit- und Speicheranforderungen handhabbar zu gestalten wurden in dieser Arbeit einige Annahmen bzw. Näherungen getroffen:

- Unabhängigkeit zwischen verschiedenen Phonemen
- Unabhängigkeit zwischen den Zuständen eines HMMs

Die erste Annahme kann für Phoneme stark diskriminativer "Klassen", z.B. Plosive und Vokale, noch als hinreichend zutreffend angenommen werden. Bei ähnlichen Phonemen, beispielsweise /a/ und /a:/, sowie bei den Einzelzuständen muss jedoch bereits von Näherungen ausgegangen werden. Der Speicherbedarf MEM reduziert sich durch diese Festlegungen auf

$$MEM = \sum_{i=1}^{N_S} (N_D^{EV}(i))^2 * 4Byte \approx N_S * (\bar{N}_D^{EV,s})^2 * 4Byte \tag{2.46}$$

Bei $N_S = 132$ und 64 Verteilungen je Zustand ergibt sich somit insgesamt $MEM = 132 * (64*42)^2 * 4Byte \approx 3.5Gb$. Die Einführung dieser Approximationen führt dazu, dass $\boldsymbol{C}_Z$ in eine Block-Diagonal Struktur unterteilt wird. Jeder Block in der Matrix $\boldsymbol{C}_Z$ entspricht dabei einem HMM-Zustand, entsprechend der Anzahl und Dimension der enthaltenen Gaussmittelpunkte.

$$\boldsymbol{C}_Z = \begin{pmatrix} d_{11} & d_{12} & & & & & & \\ d_{21} & d_{22} & & & & & & \\ & & e_{11} & e_{12} & e_{13} & & & \\ & & e_{21} & e_{22} & e_{23} & & & \cdots \\ & & e_{31} & e_{32} & e_{33} & & & \\ & & & & & f_{11} & f_{12} & \\ & & & & & f_{21} & f_{22} & \\ & & & \vdots & & & & \ddots \end{pmatrix} \tag{2.47}$$

Durch die blockweise Struktur der Matrix reduziert sich die PCA-Transformation der Gesamtmatrix auf N_S Zerlegungen der Submatrizen $\boldsymbol{C}_Z^s$. Aufgrund der mindestens quadratischen Ordnung der Algorithmen zur Matrixzerlegung kann hierdurch der Rechenaufwand auf weniger als $\frac{1}{N_S}$ des ursprünglichen reduziert werden. Darüber hinaus muss der mit MEM bezeichnete Gesamtspeicheraufwand nicht gleichzeitig bereitgestellt werden. Die zustandsweisen PCA-Zerlegungen können unabhängig voneinander, d.h. auch sequentiell, durchgeführt werden. Daher ist lediglich ein Speicheraufwand von ca. $\frac{1}{132} 3.5Gb \approx 26.5Mb$ zur Berechnung nötig. Der für die Speicherung der Matrix $\boldsymbol{C}_Z^s$ benötigte Plattenplatz kann wieder freigegeben werden, da diese nach erfolgter Berechnung der Eigenvoices nicht mehr benötigt wird. Die zur

Adaption weiterhin benötigten Eigenvoices werden jedoch in ihrer Anzahl durch den Rang der Matrix $\boldsymbol{C}_Z^s$ beschränkt. Er beträgt beim verwendeten Verbmobil-Korpus $K_{EV}^{max} = 612$. Da zur Adaption i.d.R. nur die wichtigsten Eigenvoices eingesetzt werden, kann die Anzahl deutlich weiter eingeschränkt werden, z.B. $K_{EV} = 100$. Der reine Plattenbedarf zur Speicherung der für die Adaption benötigten Parameter reduziert sich für das System mit 8500 Verteilungen somit zu $132 * (100 + 1) * 64 * 42 * 4Byte \approx 130Mb$. Dies stellt bereits einen praktisch realisierbaren Wert dar.

Die Verwendung einer globalen Matrix (alle Zustände) "impliziert" die Annahme eines gemeinsamen Unterraums, d.h. ein gemeinsamer Gewichtskoeffizient für *alle* Komponenten einer Eigenachse.

$$\boldsymbol{z} = \boldsymbol{m}_0 + \boldsymbol{E}_{EV}\boldsymbol{w} \quad \text{mit} \quad \boldsymbol{E}_{EV} = [\boldsymbol{e}_1\ \boldsymbol{e}_2\ \ldots\ \boldsymbol{e}_{K_{EV}}] \tag{2.48}$$

Bei K_{EV} verbleibenden Eigenachsen nach der Reduktion entspricht dies genau K_{EV} zu schätzenden Parametern. Man könnte diesen Ansatz als ein Koeffizienten-Tying über alle Zustände auffassen. Im Falle (quasi-)separater Eigenräume kann ein individueller Gewichtsvektor $\boldsymbol{w}^s$ für jeden Zustand s bestimmt werden, d.h.

$$\boldsymbol{z}_s = \boldsymbol{m}_0^s + \boldsymbol{E}_{EV}^s\boldsymbol{w}^s \quad \text{mit} \quad \boldsymbol{E}_{EV}^s = [\boldsymbol{e}_1^s\ \boldsymbol{e}_2^s\ \ldots\ \boldsymbol{e}_{K_{EV}}^s] \tag{2.49}$$

$\boldsymbol{E}_{EV}^s$ entspricht dabei der zum Zustand s gehörende Block der Transformationsmatrix $\boldsymbol{E}_{EV}$. Insgesamt steigt dadurch die Zahl zu schätzender Parameter auf $K_{EV} * N_S$ an. Wie durch die Annahmen festgelegt, werden die Korrelationen zwischen den Modellparametern unterschiedlicher Blöcke vernachlässigt, d.h. die entsprechenden Einträge der Matrix sind zu 0 angenommen. Daher kann für jeden Block $\boldsymbol{C}_Z^s$ der Matrix $\boldsymbol{C}_Z$ eine individuelle Eigenwertzerlegung durchgeführt werden. Allerdings kann aufgrund der individuellen Normierung und Ausrichtung der blockweisen Eigenvoices keine Rekonstruktion der globalen Eigenvoices durch direktes Zusammenfassen der "Teil"Eigenvoices erfolgen.

Die Zahl der möglichen blockweisen Eigenrichtungen wird durch die Dimension der jeweiligen Blockgrößen eingeschränkt. Bei $K_{EV} = 1..100$ Eigenstimmen stellt dies allerdings keine allzu starke Einschränkung dar, da bei 3 Normalverteilungen je Zustand bereits mehr Eigenrichtungen möglich sind ($N_D > 33$ je Verteilung angenommen).

Ein Sprechermodell lässt sich, als Punkt im reduzierten K_{EV}-dimensionalen Merkmalsraum, durch die Koordinaten $\boldsymbol{w} = [w_1, \ldots, w_{K_{EV}}]^T$ (bzw. $\boldsymbol{w}^s$ bei zustandsweisen Eigenraumkoordinaten) darstellen. Unter Berücksichtigung der Eigenvektoren $\boldsymbol{e}_k$ stellt Gl. 2.48 - nach Zerlegung in die Einzelmittelpunkte - gleichzeitig einen Punkt im N_D-dimensionalen Originalraum dar. Diesem 'Punkt' entspricht natürlich gleichzeitig der volle Parametersatz der Hidden-Markov-Modelle des Sprechers. Im Originalraum kann diese Gleichung daher als Beschränkung angesehen werden. Die Parameter können sich nicht mehr beliebig im Raum bewegen, sondern werden durch die erzwungene Lage bzw. Projektion auf die K_{EV}-Eigenachsen begrenzt.

2.5.3 Das "Maximum Likelihood Eigenspace Decomposition"-Adaptionsverfahren (MLED)

Durch die Beschränkung der Modellparameter auf den Eigenraum erfolgt eine ausgeprägte Reduktion der Parameterzahl. Unter Kenntnis des zugrundeliegenden globalen Eigenraums reduziert sich die nötige Zahl auf K_{EV}^{max} bei einem *bekannten*, d.h. bei der Generierung des Eigenraums beteiligten, Sprechermodell. Neue Sprechermodelle können jedoch in diesem Raum nicht vollständig repräsentiert werden. Bei einem robust geschätztem Modell führt eine Projektion auf den Eigenraum zu einem Verlust von Information. Wenn allerdings eine robuste Schätzung aufgrund mangelnder Trainings-/Adaptionsdaten nicht möglich ist, kann der Eigenraum als zusätzliche Apriori-Wissensquelle hinzugezogen werden. Er gibt in kompakter Form Aufschluss über den Bereich des Merkmalsraums, in dem sich die Sprechermodellparameter bewegen. Die Lage der Parameter des neuen Modells kann daher bei der Optimierung effektiv auf diesen Bereich eingeschränkt werden.

Die Parameter $\boldsymbol{w}$ ließen sich prinzipiell durch die Projektion in den Eigenraum bestimmen. Der Nachteil hierbei ist, dass dazu zuerst vollständige Sprechermodelle im Originalraum bestimmt werden müssen. Speziell bei wenig Trainings- bzw. Adaptionsdaten können die $N_D * N_{Bf}$ Parameter allerdings nicht robust genug geschätzt werden. Die nachfolgende Projektion kann die entstandenen Schätzfehler u.U. nur teilweise wieder eliminieren. Vorteilhafter wäre die Bestimmung der Gewichtsparameter $\boldsymbol{w}$ direkt im K_{EV}-dimensionalen Subraum. Die Ableitung der Gewichtskoeffizienten kann über eine Maximum-Likelihood Schätzung erfolgen [Kuh98, Kuh99, Ngu99, Kuh00]. Die Maximierung der Likelihood $p(\boldsymbol{\mathcal{X}}|\boldsymbol{s}^*, \lambda)$ erfolgt hier bezüglich der Gewichte $\boldsymbol{w}$. Abb. 2.4 zeigt dies schematisch für den Fall eines 2-dimensionalen Originalraums und unter Verwendung von $K_{EV} = 1$ Eigenvoice.

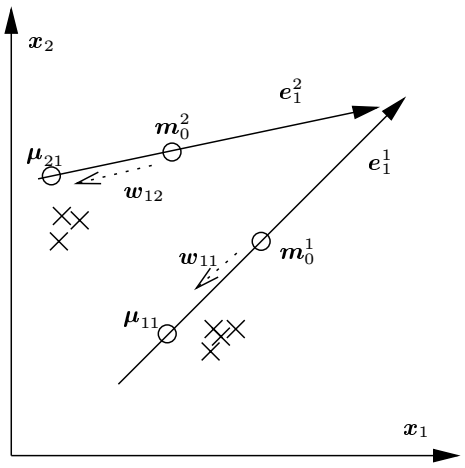

Abb. 2.4: Optimierung der Eigenvoice-Koeffizienten $\boldsymbol{w}$ durch Rückprojektion in den Originalraum.

Die Suche des optimalen Parametersatzes erfolgt nur in dem durch die Eigenachsen aufgespannten Raum. In obiger Skizze können die neuen Mittelpunkte $\boldsymbol{\mu}_{21}$ und $\boldsymbol{\mu}_{11}$ daher ($K_{EV} = 1$) nur auf der eingezeichneten 1. Eigenachse zu liegen kommen. Jedem Mittelpunkt i

entspricht immer nur ein N_D-dimensionaler Teil $\boldsymbol{e}_1^i$ der kompletten Eigenachse $\boldsymbol{e}_1 = \binom{\boldsymbol{e}_1^1}{\boldsymbol{e}_1^2}$. Die Richtung dieser Achse unterscheidet sich natürlich für verschiedene Mittelpunkte, sollte aber bei einer vorliegenden Korrelation etwa die gleiche Richtung aufweisen. Bei einer globalen Darstellung gemäß Gl. 2.39 ergibt sich für die beiden gezeichneten Mittelpunkte *ein* gemeinsamer Achsenabschnitt $w_1 = w_{11} = w_{12}$, d.h. $\boldsymbol{\mu}_{11} = \boldsymbol{m}_0^1 + w_1\boldsymbol{e}_1^1$ bzw. $\boldsymbol{\mu}_{21} = \boldsymbol{m}_0^2 + w_1\boldsymbol{e}_1^2$. Die Optimierung der Gewichte kann auch individuell erfolgen ($w_{11} \neq w_{12}$). Für die blockweise Zerlegung der Matrix $\boldsymbol{C}_Z$ bietet sich ein solches Vorgehen direkt an. Unter der Annahme, die beiden Mittelpunkte seien unterschiedlichen Zuständen, d.h. Blöcken der Matrix $\boldsymbol{C}_Z$ zugeordnet, kann eine individuelle Optimierung erfolgen. Für unterschiedliche "Teilabschnitte" einer Eigenachse ergeben sich hierdurch unterschiedliche Gewichtungsfaktoren. Die nachfolgende Ableitung der Nachschätzgleichungen, anhand des Segmental K-Means (s. Abschnitt 2.3.1) Prinzips, erfolgt für eine blockweise, d.h. zustandsspezifische Gewichtung der Eigenvoices [Fal01a]. Ein ähnlicher Ansatz zur blockweisen Einteilung und Schätzung findet sich in [Tsa01]. Die Autoren schlagen zwei unterschiedliche Vorgehensweisen zur Blockeinteilung vor. Bei der ersten werden die Blöcke an den strukturell unterschiedlichen Bestandteilen der Merkmalsvektoren (statische, Delta- und DeltaDelta-Koeffizienten) ausgerichtet. Im zweiten Ansatz gruppieren die Autoren die einzelnen Mittelpunktsvektoren anhand eines Abstandsmaßes, wobei für jede Gruppe eigene Eigenvoice-Koeffizienten ermittelt werden.

Die Maximierung der Likelihood $p(\boldsymbol{\mathcal{X}}|\boldsymbol{s}^*, \lambda)$ kann unter Zuhilfenahme der Hilfsfunktion $Q(\lambda, \bar{\lambda})$ (s. Gl. 2.13) erfolgen. Für den Fall Gauss'scher Mixturverteilungen in den einzelnen HMM-Zuständen, konnte die Hilfsfunktion, wie in Abschnitt 2.3.1 gezeigt, schließlich auf die Form in Gl. 2.16 gebracht werden. Für die Maximierung muss gelten: $\frac{\partial}{\partial\lambda}Q(\lambda, \bar{\lambda})$=0. Als zu optimierende Parameter λ sind hier die Eigenraumgewichte $\boldsymbol{w}^s = \left[w_1^s, w_2^s, \ldots, w_{K_{EV}}^s\right]^T$ zu bestimmen. Durch die Rückprojektion der Eigenraumkoordinaten in den Originalraum (Gl. 2.39) ist in direkter, $\boldsymbol{w}^s$-parametrisierter Form ein Sprechermodell gegeben, das zur Berechnung der Likelihood $p(\boldsymbol{\mathcal{X}}|\boldsymbol{s}^*, \lambda)$ der Mustervektoren verwendet werden kann. Bei der Verwendung zustandsweiser Gewichte $\boldsymbol{w}^s$ zerfällt Gl. 2.16 aufgrund der unabhängigen Mixturverteilungen in N_S unabhängig optimierbare Teilprobleme.

$$Q(\lambda, \bar{\lambda}) = \sum_{i=1}^{N_S} Q_i(\lambda, \bar{\lambda}) \tag{2.50}$$

$$\begin{aligned}
\frac{\partial Q_i}{\partial w_j^i} &= p(\boldsymbol{\mathcal{X}}|\boldsymbol{s}^*, \lambda) \sum_{\kappa=1}^{K}\sum_{t=1}^{T} \gamma_{i\kappa}\delta(s_t^* - i)\frac{\partial}{\partial w_j^i}(\log \bar{\mathcal{N}}_{i\kappa}(\boldsymbol{x}_t)) = \\
&= p(\boldsymbol{\mathcal{X}}|\boldsymbol{s}^*, \lambda) \sum_{\kappa=1}^{K}\sum_{t=1}^{T} \gamma_{i\kappa}\delta(s_t^* - i)\frac{1}{\bar{\mathcal{N}}_{i\kappa}(\boldsymbol{x}_t)}\bar{\mathcal{N}}_{i\kappa}(\boldsymbol{x}_t) \\
&\quad * \frac{\partial}{\partial w_i^j}\left(-\frac{1}{2}(\boldsymbol{x}_t - \boldsymbol{\mu}_{i\kappa})^T\boldsymbol{\Sigma}_{i\kappa}^{-1}(\boldsymbol{x}_t - \boldsymbol{\mu}_{i\kappa})\right)
\end{aligned} \tag{2.51}$$

Mit der parametrisierten Rückprojektion der Eigenspace-Koordinaten

$$\boldsymbol{\mu}_{i\kappa} = \boldsymbol{m}_0^{i\kappa} + \boldsymbol{E}^{i\kappa}\boldsymbol{w}^i = \boldsymbol{m}_0^{i\kappa} + \sum_{j=1}^{K_{EV}} w_j^i\boldsymbol{e}_j^{i\kappa} \tag{2.52}$$

ergibt sich

$$\frac{\partial Q_i}{\partial w_j^i} = p(\boldsymbol{\mathcal{X}}|\boldsymbol{s}^*,\lambda)\sum_{\kappa=1}^{K}\sum_{t=1}^{T}\gamma_{i\kappa}\delta(s_t^*-i)(\boldsymbol{e}_j^{i\kappa})^T\boldsymbol{\Sigma}_{i\kappa}^{-1}\left(\boldsymbol{x}_t-\boldsymbol{m}_0^{i\kappa}-\boldsymbol{E}^{i\kappa}\boldsymbol{w}^i\right)=0 \quad (2.53)$$

$\boldsymbol{m}_0^{i\kappa}$ und $\boldsymbol{E}^{i\kappa}=\left[\boldsymbol{e}_1^{i\kappa}\ \boldsymbol{e}_2^{i\kappa}\ \ldots\ \boldsymbol{e}_{K_{EV}}^{i\kappa}\right]$ sind die zu Verteilung κ des Zustands i gehörigen Teilkomponenten der K_{EV} zustandsspezifischen Supereigenvektoren. Sortiert man in Gl. 2.51 die von den Gewichtsvektoren $\boldsymbol{w}^i$ abhängigen Anteile auf die rechte Seite der Gleichung und die unabhängigen Anteile nach links, so ergibt sich:

$$\sum_{\kappa=1}^{K}\sum_{t=1}^{T}\gamma_{i\kappa}\delta(s_t^*-i)(\boldsymbol{e}_j^{i\kappa})^T\boldsymbol{\Sigma}_{i\kappa}^{-1}\left(\boldsymbol{x}_t-\boldsymbol{m}_0^{i\kappa}\right)=\sum_{\kappa=1}^{K}\sum_{t=1}^{T}\gamma_{i\kappa}\delta(s_t^*-i)(\boldsymbol{e}_j^{i\kappa})^T\boldsymbol{\Sigma}_{i\kappa}^{-1}\boldsymbol{E}^{i\kappa}\boldsymbol{w}^i \quad (2.54)$$

dies entspricht der linearen Gleichung $\beta_j^i=(\boldsymbol{\phi}_j^i)^T\boldsymbol{w}^i$ mit

$$\beta_j^i = \sum_{\kappa=1}^{K}\sum_{t=1}^{T}\gamma_{i\kappa}\delta(s_t^*-i)(\boldsymbol{e}_j^{i\kappa})^T\boldsymbol{\Sigma}_{i\kappa}^{-1}\left(\boldsymbol{x}_t-\boldsymbol{m}_0^{i\kappa}\right) \quad (2.55)$$

$$(\boldsymbol{\phi}_j^i)^T = \sum_{\kappa=1}^{K}\sum_{t=1}^{T}\gamma_{i\kappa}\delta(s_t^*-i)(\boldsymbol{e}_j^{i\kappa})^T\boldsymbol{\Sigma}_{i\kappa}^{-1}\boldsymbol{E}^{i\kappa} \quad (2.56)$$

Für alle K_{EV} Eigengewichte eines individuellen Zustands i entspricht dies dem linearen Gleichungssystem:

$$\boldsymbol{\beta}^i=\boldsymbol{\Phi}^i\boldsymbol{w}^i \quad (2.57)$$

mit $\boldsymbol{\Phi}^i=[\boldsymbol{\phi}_1^i\ \boldsymbol{\phi}_2^i\ \ldots\ \boldsymbol{\phi}_{K_{EV}}^i]^T$ und $\boldsymbol{\beta}^i=[\beta_1^i\ \beta_2^i\ \ldots\ \beta_{K_{EV}}^i]^T$. Die Gewichtsvektoren in einer Iteration ergeben sich daher zu: $\boldsymbol{w}^i=(\boldsymbol{\Phi}^i)^{-1}\boldsymbol{\beta}^i$. Durch Iteration der Schätzung bzw. Maximierung kann die Likelihood sukzessive erhöht werden.

2.5.4 Experimente

Abb. 2.5 zeigt für einige repräsentative Phoneme die erklärte Varianz $\sum_{k=1}^{K_{EV}}\lambda_k^s$ in Abhängigkeit von der Dimension K_{EV} des reduzierten Eigenraums (jeweils Eigenraum des mittleren Zustands s_2). Hierzu wurden für die 613 Sprecher des Verbmobil Korpus mittels MAP individuelle HMM-Phonemmodelle trainiert. Der Verlauf der akkumulierten Varianz macht deutlich, dass bereits mit nur 50 Eigenachsen ca. 50% der jeweiligen Gesamtvarianz der Sprechermodellparameter erklärt werden kann.

Einen weiteren Aspekt verdeutlicht Abbildung 2.6. Sie vergleicht die Abhängigkeit der Gesamtvarianz von MLLR(global)- und MAP-trainierten Modellen von der Anzahl der jeweils vorhandenen Trainingsvektoren je Phonem. Analog zur MAP-Anpassung lässt sich auch die MLLR-Transformation zur Ableitung der sprecherspezifischen Modelle verwenden [ChK00]. Die Verwendung einer MLLR lässt sich zwar praktisch rechtfertigen - die theoretische Rechtfertigung für den Einsatz einer gemeinsamen Regression ist jedoch fraglich. Insbesondere die Betrachtungen in Abschnitt 2.5.2.1 lassen sich mit einer MLLR-Transformation nur bedingt

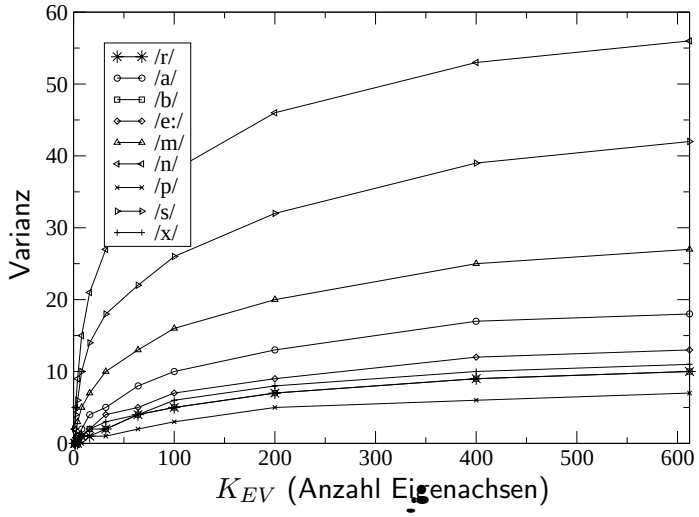

Abb. 2.5: Erklärte Varianz für verschiedene MAP-trainierte Phonemmodelle (HMMs, mittlerer Zustand) in Abhängigkeit von der Zahl der Eigenachsen K_{EV}.

nachvollziehen. Abb. 2.6 zeigt für MAP eine ausgeprägte Korrelation $\rho \approx 0.95$ zwischen Varianz und phonemweiser Framehäufigkeit. Bei Verwendung einer globalen MLLR ist die Korrelation zwischen beiden nahezu 0.

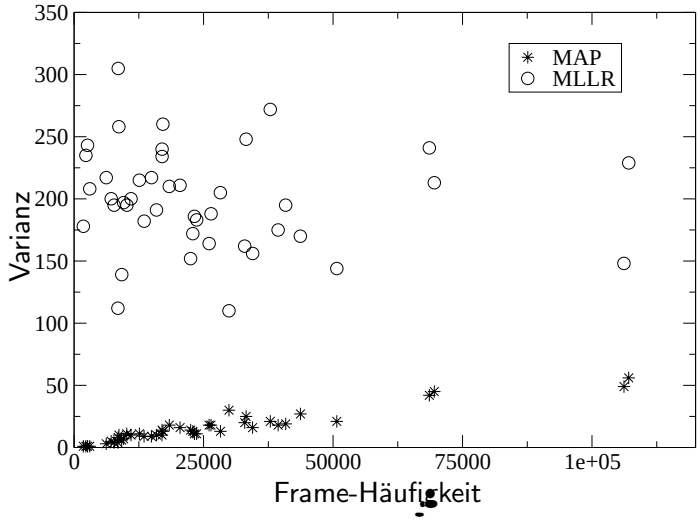

Abb. 2.6: Abhängigkeit der gesamten Inter-Sprecher Modellvarianz der Phonemmodelle (mittlerer HMM-Zustand) von der Anzahl der jeweils verfügbaren Merkmalsvektoren und dem verwendeten Trainingsverfahren.

Bei zustandsspezifischen Eigenraumgewichten (Gl. 2.49) spielt die unterschiedliche Varianz der Phonemmodelle keine Rolle, da die Koeffizienten bezüglich der zustandsweise normierten Eigenrichtungen bestimmt werden. Anders jedoch bei globalen Eigenraumkoeffizienten (Gl. 2.48): hier wird der bei ungünstiger Normierung der Eigenachsen der Koeffizientensatz

primär durch die Phoneme mit hoher Varianz, d.h. großer Häufigkeit bestimmt.

2.6 Kombination von MLED- und MLLR-Schätzung

Im folgenden wird ein Ansatz vorgestellt, bei dem eine gleichzeitige Schätzung sowohl der Eigenvoice Koeffizienten, als auch der MLLR-Transformationsmatrix erfolgt. Die Bestimmung kann dabei sowohl parallel als auch sequentiell erfolgen. Ausgangspunkt sowohl für eine MLLR- als auch eine Eigenvoice-beschränkte Adaption sind i.d.R. generische SD-Modelle. Wie gezeigt werden konnte, ist gerade bei sehr wenig Trainingsdaten die Nutzung der Apriori-Information über die Verteilung der Sprechermodelle geeignet, die Modellparameter eines neuen Sprechers einzugrenzen. Wie die Ergebnisse in Abschnitt 2.7 zeigen, kann bei zunehmenden Adaptionsmaterial die MLLR-Transformation ihre Stärken ausspielen. Bei der Verwendung generischer Ausgangsmodelle für die Transformation muss diese jedoch einen Teil ihrer Mächtigkeit darauf verwenden, die groben Änderungen zu erfassen, wie sie beispielsweise durch das Geschlecht, oder innerhalb dessen, durch den Sprechertypus verursacht werden.

Naheliegend ist also eine Kombination beider Verfahren. Die grobe Anpassung der Modellparameter kann durch die Eigenvoice-Beschränkungen erreicht werden, wohingegen die "Feinabstimmung" von der MLLR-Transformation übernommen werden kann. Die folgenden Abbildungen 2.7 und 2.8 sollen dieses Vorgehen noch einmal verdeutlichen.

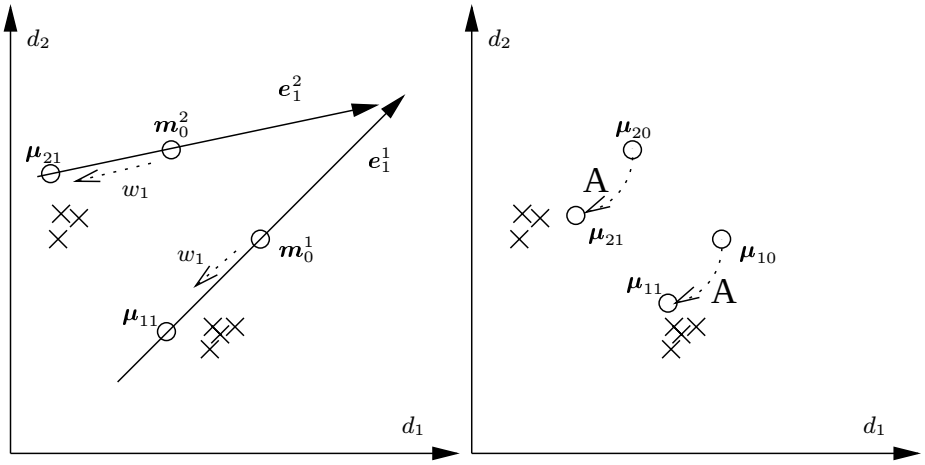

Abb. 2.7: Gegenüberstellung von Eigenvoice- und MLLR-Regression.

Die Abbildungen 2.7 stellen schematisch den Gegensatz zwischen der Verwendung von Apriori-Informationen über die Sprecherverteilung und der Anwendung einer affinen Transformation dar. Die linke Abbildung zeigt beispielhaft, wie für 2 Mittelpunkte - bei Verwendung von $K_{EV} = 1$ Eigenvoice - die möglichen adaptierten Mittelpunkte nur auf dieser Achse zum Liegen kommen können. Im allgemeinen Fall handelt es sich um einen K_{EV}-dimensionalen Subraum. Dem gegenüber steht die lineare (bzw. affine) Transformation in der rechten Abbildung. Hier werden die neuen Mittelpunkte durch eine (optionale) Verschiebung, sowie Drehung und Skalierung der alten Mittelpunkte erzeugt. Inwieweit dies möglich ist hängt

davon ab, über welche Modelle, HMM-Zustände oder Mittelpunkte hinweg die Bestimmung der jeweiligen gemeinsamen Transformationsmatrix vorgenommen wird.

Beide Adaptionsverfahren lassen sich sequentiell kombinieren. Nachteilig an einem getrennten Vorgehen ist, dass die beiden Parametersätze im Prinzip unabhängig voneinander bestimmt werden, d.h. eine nachfolgende MLLR-Transformation baut bereits auf einem fixen MLED-Ausgangsmodell auf. Speziell im Hinblick auf eine optimierte Likelihood wäre es wünschenswert, beide Ansätze in einem Schritt zu kombinieren.

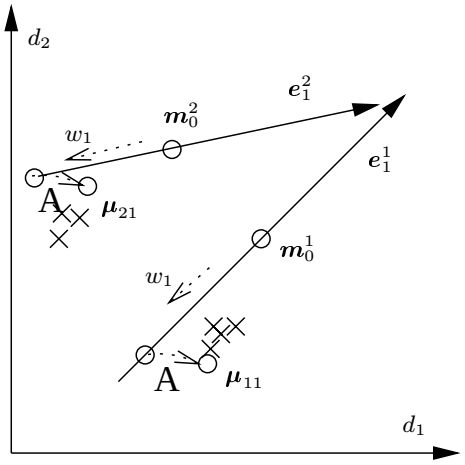

Abb. 2.8: Kombinierte Schätzung von MLLR-Transformationsmatrix und Eigenvoice-Koeffizienten.

Abb. 2.8 verdeutlicht die Idee hinter der Kombination beider Verfahren. Der Eigenvoice-Anteil lässt nur Mittelpunkte entlang der Eigenachsen zu. Durch die Kombination mit der MLLR dienen diese jedoch nur als temporäre Mittelpunkte, die auf die endgültigen Prototypen abgebildet werden. Da die Optimierung der Eigengewichte nicht unmittelbar bezüglich der Adaptionsdaten erfolgt, sondern indirekt über die zwischengeschaltete Transformation, können sich also andere Gewichte einstellen.

Wie bei der nachfolgenden Herleitung der kombinierten Schätzung erläutert, lässt sich die Betrachtungsweise, d.h. Reihenfolge, der Transformationen auch umkehren. Demnach wird durch die affine Transformation in einem ersten Schritt das Eigensystem transformiert. Anschließend kann die Position der Mixturparameter im neuen System ermittelt werden. Kernpunkt des Ansatzes ist jedoch die kombinierte Optimierung beider Parametersätze, d.h. sowohl der Transformationsmatrix, als auch der Eigengewichte. Ermöglicht wird dies durch die Integration beider Transformationen in die Likelihoodberechnung.

$$p(\boldsymbol{x}_t|s) = \sum_{k=1}^{K_{ms}} \frac{1}{\sqrt{(2\pi)^{N_D}|\boldsymbol{\Sigma}_{sk}|}} \exp\left(-\frac{1}{2}\boldsymbol{y}_{sk}^T(t)\boldsymbol{\Sigma}_{sk}^{-1}\boldsymbol{y}_{sk}(t)\right) \tag{2.58}$$

Für die MLLR gilt einerseits:

$$\boldsymbol{y}_{sk}(t) = \boldsymbol{x}_t - \boldsymbol{b}_r - \boldsymbol{A}_r\boldsymbol{\mu}_{sk} \tag{2.59}$$

Andererseits gilt für die Limitierung mittels Eigenvoices:

$$\boldsymbol{y}_{sk}(t) = \boldsymbol{x}_t - \boldsymbol{m}_0^{sk} - \boldsymbol{E}^{sk}\boldsymbol{w}^s \tag{2.60}$$

In ersterem Falle sind die Elemente der Transformationsmatrizen $\boldsymbol{A}_r$ die zu bestimmenden Parameter, wohingegen im zweitem Fall die Gewichte $\boldsymbol{w}^s$ der Eigenachsen (=Koordinaten im Eigenraum) die gesuchten Unbekannten sind. Eine Kombination der beiden Transformationen der SD-Modelle ergibt also:

$$\boldsymbol{y}_{sk}(t) = \boldsymbol{x}_t - \boldsymbol{b}_r - \boldsymbol{A}_r(\boldsymbol{m}_0^{sk} + \boldsymbol{E}^{sk}\boldsymbol{w}^s) \tag{2.61}$$

Prinzipiell könnten beide Parametersätze unabhängig, nacheinander geschätzt werden - jedoch auf Kosten einer möglichen, verbesserten Modellierung. Im folgenden soll gezeigt werden, dass beide auch näherungsweise gemeinsam bzgl. des Maximum-Likelihood Prinzips optimiert werden können. Ansatzpunkt hierzu ist wiederum die Hilfsfunktion $Q(\lambda, \bar{\lambda})$, ausgehend von Gl. 2.16. Für eine Maximierung von $p(\boldsymbol{\mathcal{X}}, \boldsymbol{s}^*|\lambda)$ mit $\lambda = \{\boldsymbol{b}_r, \boldsymbol{A}_r, \boldsymbol{w}^s\}$ muss wiederum gelten $\frac{\partial}{\partial\lambda}Q(\lambda, \bar{\lambda}) = 0$. Die partielle Ableitung erfolgt hier sowohl bzgl. der Elemente der Regressionsmatrix a_{ij} als auch bzgl. der Eigenvoice-Gewichte w_j^s.

Da es sich beim der Maximum-Likelihood Schätzung um ein iteratives Verfahren handelt, kann bei der partiellen Ableitung der jeweils andere Parametersatz näherungsweise als konstant im Arbeitspunkt betrachtet werden. Gegenseitige Abhängigkeiten werden hierdurch innerhalb einer Iteration vernachlässigt und die gesuchten Parameter können unabhängig voneinander, mittels der leicht modifizierten, ursprünglichen Gleichungen 2.38 und 2.57 bestimmt werden. Zur Bestimmung der Eigenvoice-Gewichte kann daher in Gleichung 2.61 die Transformationsmatrix nach "links" geschoben werden und direkt mit den Eigenvektoren multipliziert werden. Dadurch ergeben sich die transformierten Eigenvektoren $\hat{\boldsymbol{e}}_j^{sk}$ mit

$$\hat{\boldsymbol{e}}_j^{sk} = \boldsymbol{A}_r\boldsymbol{e}_j^{sk} \tag{2.62}$$

Ersetzt man $\hat{\boldsymbol{e}}_j^{sk}$ in Gl. 2.56 bzw. 2.57 so können die Lösungen der partiellen Ableitung mittels dieser Gleichung ermittelt werden. Werden andererseits die Eigengewichte in 2.61 als konstant angesehen, dann ergeben sich die veränderten Mittelpunkte

$$\hat{\boldsymbol{\mu}}_{sk} = \boldsymbol{m}_0^{sk} + \boldsymbol{E}^{sk}\boldsymbol{w}^s \tag{2.63}$$

Werden diese wiederum in Gl. 2.38 für $\boldsymbol{\mu}_{sk}$ eingesetzt, so können wiederum die ursprünglichen Gleichungen zur Bestimmung der Matrixelemente $a_{ij}(\kappa + 1)$ herangezogen werden. Im darauffolgenden Iterationsschritt können nun die neu geschätzten Parameter eingesetzt werden. Durch die fortgesetzte Iteration nähern sich die Parameter der Maximum-Likelihood Lösung. Im Prinzip handelt es sich bei diesem Ansatz um eine iterative sequentielle Kombination der beiden Verfahren.

2.7 Vergleichende Adaptionsexperimente

Um die Leistung der verschiedenen Adaptionsalgorithmen bewerten und vergleichen zu können, wurden Adaptionsexperimente durchgeführt. Das experimentelle Umfeld beinhaltete dabei 40 Sprecher, für die jeweils 2 bzw. 10 Adaptionsäußerungen vorlagen (s. Kap. 1.5). Die für die Adaption nötige phonetische Verschriftung und Segmentierung der Äußerungen wurde mittels der generischen SI-Monophonmodelle aus Tab. 5.1 erzeugt. Von Interesse ist in diesem Zusammenhang insbesondere das Verhalten bei unüberwachter Anpassung. Aus diesem Grund wurden 2 Forced-Viterbi-Segmentierungen erzeugt. Grundlage der ersten ist die korrekte Wortfolge der Äußerungen, die als bekannt vorausgesetzt wurde. Für die zweite Segmentierung wurde die Wortfolge mittels einer vollständigen, automatischen Erkennung erzeugt. Tab. 2.1 zeigt die ermittelten Wortfehlerraten (WER, engl. 'word error rate'). Bei der MLED wurden zustandsweise Koeffizienten gemäß Gl. 2.50 bzw. 2.57 berechnet.

Adaption	Trainingsalg.	K_R	K^s_{EV}	#Turns	#Param.	WER [%]
SI-Basissystem	ML	-	-	-	722000	38,7
überwacht	MLLR	1	-	10	1806	34.7
		3	-	10	5418	34.3
		5	-	10	9030	34.1
	MLED	-	4	10	528	36.7
		-	8	10	1056	35.7
		-	16	10	2112	35.4
		-	50	10	6600	37.4
	MLED+MLLR (komb.)	1	4	10	2334	34.6
		1	16	10	3918	34.5
	MLED+MLLR	1	16	10	3918	34.6
	MLLR	1	-	2	1806	37.9
	MLED	-	4	2	528	39.1
		-	16	2	2112	41.7
	MLED+MLLR (komb.)	1	4	2	2334	39.4
unüberwacht	MLLR	1	-	10	1806	35.5
	MLED	-	4	10	528	43.5
		-	16	10	2112	43.3
	MLLR	1	-	2	1806	38.4
	MLED	-	4	2	528	68.7
		-	16	2	2112	68.7

Tab. 2.1: Wortfehlerrate nach Adaption mit 2 bzw. 10 Äußerungen.

Beide Verfahren ermöglichen bei überwachter Adaption selbst bei wenig Adaptionsdaten eine Verringerung der WER von bis zu 3.6% absolut. Im Vergleich zur Gesamtzahl der Systemparameter ist die Zahl der zu schätzenden Unbekannten bei beiden Ansätzen erheblich niedriger. Die Erkennungsexperimente machen aber deutlich, dass die MLLR-Anpassung im Vergleich zu einer rein zustandsweisen MLED-Anpassung - bei dieser Menge von Trai-

ningsdaten - robuster ist. Aufgrund des aufgegebenen Zustandstyings wirken sich bei der MLED-Adaption, speziell bei unüberwachten Training, Fehlsegmentierungen stärker aus. Die Zustands-Fehlzuweisungen der Viterbi-Segmentierung führen somit zu einem drastischen Anstieg der Fehlerrate. Setzt man die Wortfolge allerdings als bekannt voraus, so kann mit dem EV-Ansatz nahezu die Performanz der MLLR-Anpassung erzielt werden. Die iterative Kombination beider Verfahren erlaubt eine geringfügige weitere Reduktion der Fehlerrate ($K_R = 1$, $K_{EV}^s = 4$ bzw. 16).

Kapitel 3

Untersuchungen zur Sprechgeschwindigkeit

3.1 Einführung und Motivation

Eine der primären Zielsetzungen dieser Arbeit ist die Generierung von robusten, akustisch-phonetischen HMM-Modellen für die automatische Spracherkennung. Mittels dieser sollen störende Abweichungen, bedingt durch Variationen der Sprechgeschwindigkeit, sowie Wechsel des Sprechers, erfasst und ausgeglichen werden können. Große Trainingskorpora enthalten i.d.R. eine beträchtliche Varianz, sowohl hinsichtlich der enthaltenen Sprecher, als auch hinsichtlich deren Sprechstils. Es lassen sich also Beispiele in der Datenbasis finden, die der aktuellen Spracheingabesituation (Sprecher und Sprechgeschwindigkeit) ähnlich sind. Für die Erzeugung robuster Modelle müssen allerdings ausreichend Beispieldaten vorliegen. Da dies jedoch häufig nicht der Fall ist, muss ein Mittelweg zwischen Genauigkeit und Menge der Beispieldaten beschritten werden. Als zentrale Maßnahme hierzu bietet sich die Bildung charakteristischer Modellgruppen an.

Grundlage der Ausbildung von Gruppen stellt eine vorausgehende Analyse der Auswirkung genannter Variationen auf das Erkennungssystem dar. Als vorrangig betroffene Komponenten eines solchen Systems wird die stochastische Modellierung im Zusammenspiel mit der akustischen Merkmalsbildung (s. Kap 1.4), wie sie bei aktuellen Spracherkennungssystemen eingesetzt wird, näher untersucht. In diesem Kapitel liegt der Schwerpunkt auf der Analyse der sprechgeschwindigkeitsbedingten Auswirkungen. Die akustisch-phonetischen Veränderungen, die durch die Variation der Sprechgeschwindigkeit verursacht werden, schlagen sich in gewissem Umfang in den berechneten Merkmalsvektoren nieder. Die Mustervektoren stellen jedoch die Grundlage der statistischen Modellierung dar. Ein Schwerpunkt dieses Abschnitts ist daher die Untersuchung, in welchem Maße Mustervektoren beeinflusst werden bzw. aus den Abweichungen zwischen Modell und aktuellen Mustervektoren quantitative Rückschlüsse auf die vorliegende Sprechrate gezogen werden können. Ein weiterer Kernpunkt ist die Ausarbeitung von Kriterien zur Qualitätsbewertung der Modellierung bei gegebener Vorverarbeitung.

3.2 Stand der Technik

3.2.1 Basismaße der Sprechgeschwindigkeit

In der Literatur wurden verschiedene Methoden und Maße zur Erfassung der Sprechgeschwindigkeit (ROS, engl. 'Rate of Speech') vorgeschlagen. Als Gemeinsamkeit der meisten Vorschläge kann die Festlegung der Sprechgeschwindigkeit als 'Rate' gesehen werden. Die Rate ergibt sich im Prinzip als Quotient der innerhalb eines Beobachtungszeitraums gezählten linguistischen Einheiten N_{le} und dessen Länge T_{Beob}.

$$ROS = \frac{N_{le}}{T_{Beob}} = \frac{N_{le}}{\sum_{N_{le}} T_i} \tag{3.1}$$

Die Festlegung des Beobachtungszeitraums muss sich allerdings nicht auf die feste Lautgrenzen beschränken, wie in den Abschnitten 3.2.3 und 3.2.4 gezeigt wird. In Abwandlung der Berechnungsmethode in Gl. 3.1 (IMD, engl. 'inverse mean duration') wurden u.a. in [Mir96, Mar97] Alternativansätze vorgeschlagen, beispielsweise:

$$ROS = \frac{\sum_{N_{le}} \frac{1}{T_i}}{N_{le}} \tag{3.2}$$

Die Berechnung anhand Gl. 3.2 (MR, engl. 'mean rate') mittelt über die individuellen 'Raten' $\frac{1}{T_i}$ der einzelnen linguistischen Einheiten innerhalb des Beobachtungszeitraums. Obwohl sich die Berechnungsmethode unterscheidet, bleibt jedoch die qualitative Auffassung als Rate, d.h. als Anzahl pro Zeiteinheit, gegeben. Die Methoden zur Sprechgeschwindigkeitsbestimmung lassen sich anhand verschiedener Kriterien unterscheiden, wobei an erster Stelle sicherlich die linguistische Einheit, deren Abfolge gezählt wird, steht.

- Phon- bzw. Phonemrate
- Silben-, Vokalrate
- Wortrate

Die Phon- bzw. Phonemrate ROS_{Ph} kann als direktes Maß für die artikulatorische Sprechgeschwindigkeit gesehen werden, da sie unmittelbar die Anzahl der geäußerten Laute pro Zeiteinheit wiedergibt. Dieses Kriterium wurde vielfach untersucht und eingesetzt, beispielsweise in [Mir95, Mir96, Sie95, Mor97, Mar98]. Die Bestimmung der lexikalischen Phonemrate erfolgt dabei meist anhand einer phonetischen Forced-Viterbi Segmentierung der sprachlichen Äußerung. Die Segmentierung erlaubt die Bestimmung der Anzahl $N_{le} = N_{Ph}$ sowie der Dauer T_i der sprachlichen Einheiten, die für die Berechnung der Phonemrate notwendig ist. Erfolgt die Bestimmung der Phonemrate global für eine komplette Äußerung, so ist eine Segmentierung nicht nötig, sofern die Gesamtlänge sowie die Anzahl der enthaltenen Einheiten bekannt ist.

Pfitzinger [Pfi96] unterscheidet hier zwischen 'brutto' (engl. 'gross') und 'netto' (engl. 'net') Sprechgeschwindigkeit. Erstere ergibt sich, wenn die Anzahl der Laute aus der kanonischen (Duden) Lexikonumschrift der enthaltenen Wörter ermittelt wird. Gerade bei spontansprachlichen Äußerungen, oder bei höherer Sprechgeschwindigkeit nimmt jedoch bei Sprechern die Tendenz zu, Laute auszulassen (Elisionen) oder in einen Laut 'zusammenzufassen'

(Assimilation). Ein weiterer, mit beiden Effekten direkt verbundener Aspekt, der häufig stark sprecher- und regionalspezifisch ausgeprägt ist, ist die Verwendung von Aussprachevarianten. Aus sprachökonomischen Gründen fallen diese Varianten oft kürzer als die zugehörige kanonische Aussprache aus. Dementsprechend sinkt die Anzahl der real geäußerten phonetischen Einheiten im Beobachtungszeitraum. Ein konkretes Beispiel hierfür wäre die im Deutschen häufig vorkommende Aussprachevariante /h a m/ des Worts 'haben' (kanonisch: /h a b @ n/). Im kanonischen Fall würden 5 Lauteinheiten angenommen, wohingegen real nur 3 Einheiten geäußert werden. Wird die Gesamtzahl der Laute im Beobachtungszeitraum anhand der wirklich gesprochenen Laute bestimmt, so ergibt sich daraus die Nettorate. Experimentell lässt sich diese ermitteln, indem bei einer phonetischen Forced-Viterbi Segmentierung auch Aussprachevarianten als Wortalternativen zugelassen werden. Im Laufe dieser Arbeit wurde schwerpunktmäßig mit der Phonemrate, insbesondere der Nettophonemrate, als Basismaß gearbeitet.

Zur Bestimmung der lexikalischen Nettosprechrate ist die Kenntnis der in einem Satz verwendeten Aussprachevarianten notwendig. Zu diesem Zweck wurde in dieser Arbeit das von Pfau [Pfa00b] beschriebene, mehrstufige Vorgehen aufgegriffen und implementiert, das die Bestimmung von Aussprachevarianten bzw. das Training passender akustischer Modelle zum Ziel hat. Im ersten Schritt wurden hierzu Worthypothesengraphen generiert, die parallel für jedes Wort eines Satzes, dessen häufigste Aussprachevarianten enthalten. Initiale akustische Aussprachevarianten(HMM)-Modelle wurden anhand des handsegmentierten Korpus aus Kiel trainiert. Mittels einer Viterbi-Suche wurde die bezüglich der Sätze und der eingesetzten Modelle wahrscheinlichste Abfolge von Varianten ermittelt. Mit dem auf diese Weise variantensegmentierten Verbmobil Korpus wurden robustere, varianten-basierte akustische Modelle trainiert, die die Grundlage zur Erzeugung der erforderlichen 'Netto'Phonemsegmentierungen bildeten.

Weitere lexikalisch motivierte Maßeinheiten sind die Wortrate, sowie die Silbenrate. Die Wortrate spielt (bezüglich Spracherkennungssystemen) eine eher untergeordnete Rolle, da sie weniger aussagekräftig ist. So fließt bei der Wortrate indirekt die Länge der Wörter mit ein, wie die beiden bekannten Beispiele "How to wreck a nice beach" und "How to recognize speech" sehr schön zeigen. Beide Beispiele enthalten annähernd die gleiche Anzahl an Phonemen, jedoch eine stark unterschiedliche Anzahl von Wörtern. Die Wortrate ist damit primär durch den lexikalischen Inhalt einer Äußerung bestimmt. Speziell Siegler und Stern konnten in [Sie95] einen eindeutigeren Zusammenhang zwischen Wortfehlerrate und Phonrate feststellen als zwischen WER und Wortrate.

Aussagekräftiger als die Wortrate ist die Silbenrate. Für betonungszählende Sprachen, wie Deutsch und Englisch, bestimmt der annähernd konstante zeitliche Abstand zwischen betonten Silben maßgeblich den Sprachrhythmus. Die Silbenrate gibt daher im Vergleich zur Wortrate genaueren Aufschluss über die artikulatorische Sprechgeschwindigkeit. Darüber hinaus sind für die genannten Sprachen die Silbenkerne meist mit Vokalen identisch [Rus94, Pfa98a, Pfa00b]. Die Bestimmung der Vokalrate ist in diesem Fall gleichbedeutend mit der Bestimmung der Silbenrate. Da jedoch die Gesamtzahl der Vokale in einem Satz kleiner ist

als die Gesamtzahl der Phoneme und damit stärker vom phonetischen Inhalt abhängt, ist eine robuste Schätzung für sehr kurze Beobachtungszeiträume im Vergleich zur Phonemrate deutlich schwieriger.

3.2.2 Verfahren zur Sprechgeschwindigkeitsbestimmung

Bei den Ansätzen zur Bestimmung der Sprechgeschwindigkeit kann in direkte und indirekte Ansätze unterschieden werden. Direkte Ansätze zielen unmittelbar darauf ab, ein zugrundeliegendes, linguistisches Basismaß oder dessen Abfolge zu bestimmen. Beispiele hierfür sind der von Pfau in [Pfa98a, Pfa00b] vorgestellte Vokalkerndetektor zur Schätzung der Vokalrate. Grundlage dieses Ansatzes ist ein modifiziertes Lautheitsmaß zur Anzeige von Vokalkernen. Über die Anzahl von Vokalkernen innerhalb des Beobachtungszeitraums kann (nach Gl. 3.1) unmittelbar auf die Vokalrate geschlossen werden. Da für die deutsche Sprache der Vokalkern meist mit dem Silbenkern zusammenfällt, kann dieser Ansatz auch zur Anzeige der Silbenrate eingesetzt werden.

Ebenfalls als direktes Verfahren kann das von Verhasselt in [Ver96] präsentierte Vorgehen bezeichnet werden. Er verwendete ein neuronales Netz (NN) mit 11 versteckten Knoten und 50 Eingängen zur Ermittlung der Phonemgrenzen. Das Multi Layer Perzeptron (MLP) wird an seinen Eingängen mit dem Spektrum und zusätzlichen Funktionen zur Erfassung von Änderungen der Energie sowie des Spektrums beaufschlagt. Der zu lernende Zielwert des NN wird zu 1 gesetzt, falls für den fraglichen Frame eine Phonemgrenze vorliegt - ansonsten zu 0. Die Abweichung zwischen geschätzter und Referenzphonemrate lag bei diesem Ansatz im Mittel unter 20%.

Das naheliegendste direkte Vorgehen zur Bestimmung sowohl der Silben- als auch der Phonemrate ist die Verwendung einer Erkennerhypothese, bzw. deren Phonemfolge [Sie95, Mir96]. Falls jedoch nur die reine Hypothese, ohne weitere Informationen über die Aussprache zur Verfügung steht, so kann daraus lediglich die Bruttorate bestimmt werden. Bei fehlenden Zeit- bzw. Segmentierungsmarken kann darüber hinaus nur auf die globale, satzweise Sprechgeschwindigkeit geschlossen werden.

Indirekte, d.h. mittelbare Ansätze zielen auf die Bestimmung eines Stellvertretermaßes ab, das in irgendeiner Form durch die Sprechgeschwindigkeit beeinflusst oder von dieser als abhängig angesehen wird. Die Qualität der Repräsentation wird meist anhand des Korrelationskoeffizienten zwischen einem linguistisch-lexikalischen Basismaß und der Schätzung bewertet. Beispiele hierfür sind die von Morgan et al. entwickelten Merkmale 'enrate' [Mor97] oder 'mrate' [Mor98]. Enrate wurde von Morgan definiert als der spektrale Mittelwert der zeitlichen Energiehüllkurve. Sie berechnet sich aus der DFT $Y(k) = DFT(w(n)x(n))$ des mit $w(n)$ gefensterten ($1-2s$ Fensterbreite), gleichgerichteten und tiefpassgefilterten ($f_g = 16\text{Hz}$) Zeitsignals $x(n)$.

$$enrate = \frac{\sum_{k=s}^{K} k|Y(k)|^2}{\sum_{k=s}^{K} |Y(k)|^2} \tag{3.3}$$

Dieses "künstliche" Maß erfasst auf diese Weise die Frequenz ($\hat{=}$ Rate) mit der energie-

reiche Spracheinheiten aufeinanderfolgen. Die Korrelation von enrate zu lexikalischer Phonemrate wurde zu 0.5 ermittelt, zur Silbenrate zu 0.4. In [Mor98] erweiterte Morgan das Maß zur sogenannten "mrate" (engl.: 'multiple rate estimators'). Sie ergibt sich im Prinzip aus der Kombination von 3 parallelen Auswertungsvarianten des enrate-Basisansatzes. Die ursprüngliche, bei enrate eingesetzte Auswertungsroutine verwendete das spektrale Moment zur Bestimmung der Abfolgefrequenz. In einfacher Abwandlung hiervon wurde von Morgan ein Peak-Zähler eingesetzt, der die Frequenz aus der Anzahl der Energiespitzen innerhalb des Beobachtungsfensters bestimmt. Der dritte Auswertungsstrang setzt auf einer Teilbandzerlegung des Sprachsignals auf. Mittels 4 Bandpässen wird das Sprachsignal zu Beginn in Teilbänder zerlegt. In jedem der Teilbänder wird individuell die Energiehüllkurve bestimmt und diese, zwischen den Bändern, durch eine punktweise Kreuzkorrelation miteinander verknüpft. Die Abfolgefrequenz wird ebenfalls durch Zählen der Maxima in der Korrelationskurve errechnet. Die letztendliche, kombinierte mrate ergibt sich durch eine Mittelwertbildung aller 3 Auswertungsstränge. Morgan spricht von einer, gegenüber enrate, deutlich gestiegenen Korrelation (bis 0.67) mit der lexikalischen Silbenrate.

Eine ganze Reihe indirekter Maße wurde von Samudravijaya et al. in [Sam98] präsentiert. Die Autoren versuchen in ihrer Arbeit die Sprechrate anhand von Teilkomponenten der in der eigentlichen Spracherkennung verwendeten Mustervektoren abzuschätzen. Vorgeschlagen wurden insbesondere MTD (engl.: 'Mean Transition Duration') als Mittelwert der cepstralen Differenzen, sowie, abgeleitet aus MTD, MTA (engl.: 'Mean Transition Area'). MTA ist ein Maß für lokale Veränderung und wurde definiert als die Fläche unter den Teilen der MTD-Kurve, an denen diese einen Schwellwert überschreitet. Als weiteres Kriterium wurde von den Autoren die Abfolge stimmhafter zu stimmloser Segmente (VSR, engl. 'Voice Switching Rate') untersucht. Die Bewertung der Maße erfolgte anhand ihrer Korrelation zu Wort- und Phonrate. Bezüglich der Wortrate bewegt sich der Korrelationskoeffizient der unterschiedlichen Kriterien zwischen 0.04 und 0.32, bezüglich der Phonrate zwischen 0.19 und 0.42.

3.2.3 Festlegung des Beobachtungszeitraums: lokale vs. globale Messung

Ein konzeptionell weitreichenderer Unterschied ergibt sich durch die Behandlung und Festlegung des Beobachtungszeitraums. Pfitzinger [Pfi96] unterscheidet hier begrifflich zwischen 'globaler' und 'lokaler' Sprechgeschwindigkeit. Seitens der reinen Berechnung sind beide Konzeptionen identisch, da in beiden Fällen die Sprechrate aus der Lautanzahl im Beobachtungszeitraum und dessen Länge berechnet werden kann (Gl. 3.1 und 3.2). Der qualitative Unterschied besteht jedoch darin, dass bei einer globalen Messung *ein Wert* für einen tendenziell längeren Beobachtungszeitraum wie einen Satz, oder einen Spurt, bestimmt wird. Bei lokaler Betrachtung hingegen liegt der Beobachtungszeitraum nur im Bereich 500ms [Pfi96] bis ca. 2s [Mor98]. Dies drückt bereits aus, dass ein globales Maß nur einen recht groben Aufschluss über das Sprechgeschwindigkeitsverhalten *während einer Äußerung* erlaubt. Dies gilt insbesondere bei sehr langen, spontansprachlichen Äußerungen (>10s).

Einen Zwischenschritt hierbei stellt die Unterteilung eines Satzes in 'Spurts' (engl.) [Mor98, Pfa00b] (auch als 'Runs' bezeichnet) dar. Spurts sind Teilabschnitte der Gesamtäußerung, die

eine einigermaßen konstante Sprechgeschwindigkeit aufweisen *sollten*. Das Problem bei der Einteilung ist jedoch, dass diese anhand der dazwischenliegenden Pausen und nicht anhand der spurtinternen Sprechgeschwindigkeitsvariation festgelegt werden. Der Gedanke hierbei ist, dass nach Pausen, die eine bestimmte Schwellenlänge überschreiten, im nachfolgenden Spurt andere Sprechgeschwindigkeitsverhältnisse anzutreffen sein können. So fand beispielsweise Pfau [Pfa00b] bei seiner Spurteinteilung des Verbmobil Korpus eine Dauerverteilung der entstandenen Segmente, die von 25 bis 1434 Frames, im Mittel 220, reichte. Die Verteilung wirft die Frage auf, ob insbesondere für längere Spurts (> 2s) die Konstanz der Sprechgeschwindigkeit gewährleistet ist.

3.2.4 Lokale Sprechgeschwindigkeit

Das Konzept der lokalen Sprechgeschwindigkeit (LSR, engl. 'Local Speech Rate') wurde u.a. von Martinez [Mar98] verwendet, um Änderungen der Lautdauer bzw. im Intonationsverlauf zu untersuchen. Die LSR beschreibt die lokale Sprechgeschwindigkeit innerhalb eines kurzen Beobachtungszeitraumes, in der Größenordnung 0.5 bis 2.0s. Im Blickpunkt steht in diesem Zusammenhang der Verlauf bzw. die Veränderung der LSR über einen längeren Zeitraum hinweg. Der Verlauf lässt sich durch eine (meist überlappende) Verschiebung des Beobachtungsfensters ermitteln. Allerdings existieren in der Literatur unterschiedliche Auffassungen über die Breite, sowie den Vorschub eines geeigneten Fensters: Martinez et al. [Mar98] verwendeten ein Fenster mit dynamischer Framebreite. Dieses war so ausgelegt, dass für eine Schätzung immer eine konstante Anzahl von Phonemen abgedeckt sein muss. Dadurch beträgt der Vorschub der Messungen minimal ein Phonem.

Einen etwas anderen Ansatz verfolgte Pfitzinger in [Pfi96]. Er hielt die Breite des Schätzfensters konstant zu 500ms. Die Phoneme, die nur teilweise abgedeckt werden, werden anhand des erfassten Anteils, gemessen an ihrer Gesamtlänge, berücksichtigt. Der Vorteil dieses Ansatzes kann darin gesehen werden, dass im Prinzip für jeden Frame einer Äußerung eine Bestimmung der LSR ermöglicht wird. Dadurch ergibt sich ein gleitender Verlauf der LSR. Ein ähnliches Vorgehen findet sich bei Morgan [Mor98]. Im Gegensatz zu Pfitzinger bevorzugt Morgan jedoch ein 2s-Fenster, das im $10ms$-Rhythmus, überlappend weitergeschoben wird. Auf Satzebene betrachtet, kann somit ein globales Maß in gewisser Weise als Mittelung der lokalen Sprechgeschwindigkeit (bzw. deren Verlauf) interpretiert werden.

Die LSR ergibt sich aus der Anzahl der beobachteten Laute innerhalb des Beobachtungszeitraums und dessen Dauer (Gl. 3.1). In Abwandlung hiervon zog Martinez jedoch die in [Mar97] gegebene Gleichung zur Sprechgeschwindigkeitsbestimmung heran, die die mittleren Lautdauern berücksichtigt. Ausgehend von Gl. 3.1 wird im folgenden das Vorgehen nach Pfitzinger aufgegriffen.

Durch den Frame-weisen, zeitlichen Vorschub werden an den Rändern des Fensters (links und rechts) 2 Phoneme u.U. nur angeschnitten (s. Abb. 3.1). Bei der Berechnung der LSR werden diese nur entsprechend des angeschnittenen Anteils berücksichtigt.

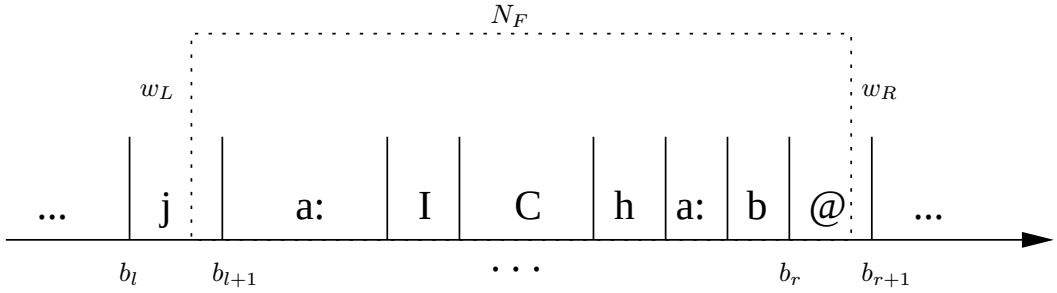

Abb. 3.1: Bestimmung der lokalen Sprechgeschwindigkeit innerhalb eines Fensters konstanter Breite.

$$N_F = b_{l+1} - w_L(n) + w_R(n) - b_r + \sum_{i=l+1}^{r-1} (b_{i+1} - b_i) \tag{3.4}$$

N_F bezeichnet die gesamte Fensterbreite, Segmentgrenzen werden durch b_i angegeben. Die Dauer eines einzelnen Phonems i in Frames ergibt sich zu: $b_{i+1} - b_i$. Bei der Bestimmung der Anzahl der durch das Fenster überstrichenen Phoneme $N_{Ph}(n)$ werden die beiden angeschnittenen Laute nur anteilig der Gesamtlänge berücksichtigt. Hierzu sind die aktuellen linken $w_L(n) = n - \frac{N_F}{2}$ bzw. rechten $w_R(n) = n + \frac{N_F}{2} - 1$ (bei geradzahligem N_F) Grenzpunkte der Fensters nötig.

$$N_{Ph}(n) = \frac{b_{l+1} - w_L(n)}{b_{l+1} - b_l} + \frac{w_R(n) - b_r}{b_{r+1} - b_r} + r - l - 1 \tag{3.5}$$

r und l geben den Index des jeweils ersten und letzten (angeschnittenen) Phonems innerhalb des Fensters an. Im folgenden wird eine Fensterbreite von $N_F = 100$ Frames zugrundegelegt. Die lokale Sprechgeschwindigkeit ergibt sich aus N_F und $N_{Ph}(n)$ nach Gl. 3.1 zu

$$LSR(n) = \frac{N_{Ph}}{N_F} = \frac{N_{Ph}(n)}{100}[Phone/Frame] \tag{3.6}$$

oder bei zugrundeliegendem Frameshift (Abstand der Kurzzeitspektren) von 10ms:

$$LSR(n) = \frac{N_{Ph}}{N_F * 10ms} = \frac{N_{Ph}(n)}{100 * 10ms} = N_{Ph}(n)[Phone/s] \tag{3.7}$$

Wie der LSR-Verlauf der Beispieläußerung (Abb. 3.2, Turn g091a000 aus Verbmobil CD1: *"Schönen guten Tag Frau Schindel ich möchte mit Ihnen gerne einige Arbeitssitzungen abmachen und zwar <NIB> zwei zweitägige und eine eintägige <NIB> und ich würde gerne anfangen mit der zweitägigen <NIB> wie wäre es denn wie würde Ihnen denn der Termin passen am Mittwoch den zehnten und am Donnerstag den elften November"*) bereits erkennen lässt, kommt es im Laufe einer Äußerung zu starken Änderungen der Sprechgeschwindigkeit. Die Wahrscheinlichkeit hierfür steigt mit der Länge der Redeflusses. Gerade bei sehr kurzen Sätzen (z.B.: "Guten Tag!") ist die Variation der LSR dagegen eher gering. Andererseits zeigt der Beispielverlauf bereits, dass die Änderung der Sprechgeschwindigkeit recht hohe Werte annehmen kann. Die Beschleunigung erreicht hierbei Werte von bis zu $15[Phone/s^2]$.

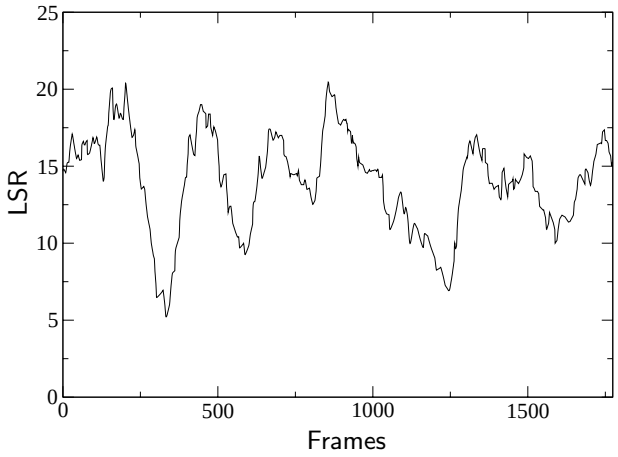

Abb. 3.2: Verlauf der lokalen Sprechgeschwindigkeit LSR auf einer typischen Beispieläußerung. (Turn g091a000 aus Verbmobil CD1).

Bei diesen Beschleunigungswerten kann sich die lokale Sprechgeschwindigkeit innerhalb von nur ca. $500ms \hat{=} 50Frames$ von langsamer zu schneller Sprechweise ändern. Speziell für spontansprachliche Äußerungen, wie sie im untersuchten Verbmobil Korpus vorliegen, kann daher davon ausgegangen werden, dass die Annahme einer konstanten Sprechgeschwindigkeit innerhalb eines Spurts nur bei sehr kurzen Spurts zutreffend ist.

3.3 Bestimmung der Spurt-weisen Sprechgeschwindigkeit

3.3.1 Kategorie-weise Sprechgeschwindigkeit

Bei der Erkennung fließender Sprache stellt die variierende Sprechgeschwindigkeit ein nicht zu vernachlässigendes Problem dar. Insbesondere bei schneller Sprache sinkt die Erkennungsleistung deutlich. Gezeigt wurde dies u.a. in [Mor97, Mar98, Pfa98b, Pfa00b, Ric99, Fal99, Fal00a, Wre01]. Von einigen der Autoren wurde die Möglichkeit untersucht, die Erkennungsrate durch die Verwendung sprechgeschwindigkeitsspezifischer Modelle zu verbessern. So verglich Pfau in [Pfa98b] das Training mittels des ML- und des MAP-Trainingsalgorithmus. In eine andere Richtung gingen die Untersuchungen in [Fal99]. Hier wurde versucht durch einen modifizierten Clusteralgorithmus die Anzahl der verwendeten Normalverteilungen für schnelle Sprache zu optimieren. Wrede andererseits verwendete in [Wre01] ein Erkennersystem, das mit semikontinuierlichen HMM (SCHMMs, engl.: 'Semi Continuous HMMs') arbeitet. Die bei wenig Trainingsdaten im Vergleich zu kontinuierlichen Modellen (CDHMMs, engl.: 'Continuous Density HMMs'), robusteren SCHMMs erlaubten ihr das Training von mehr als 3 Sprechgeschwindigkeitsklassen.

Allen Ansätzen ist gemein, dass durch das Training bzw. den Einsatz von sprechgeschwindigkeitskategoriespezifischen HMM-Modellen eine Verringerung der Wortfehlerrate erzielt werden konnte. Dies zeigt, dass die akustischen Modelle der jeweiligen Kategorie besser an die 'spektralen' bzw. temporalen Eigenheiten bei der jeweiligen Sprechgeschwindigkeit angeglichen sind. Im Umkehrschluss lässt sich aus der verringerten Wortfehlerrate der

angepassten Modelle folgern, dass sich die modellierten Spektren in Abhängigkeit von der Sprechgeschwindigkeit unterscheiden müssen. *Von Interesse ist daher die Frage, ob sich durch die kategorieweise Modellierung der sprechgeschwindigkeitsbedingten Abweichung quantitative Rückschlüsse auf die Sprechgeschwindigkeit treffen lassen.*

Die im vorausgegangenen Abschnitt 3.2.2 beschriebenen Verfahren konzentrieren sich auf die Bestimmung der Sprechgeschwindigkeit anhand zusätzlicher Merkmale, die speziell für diesen Zweck konzipiert wurden. Im Gegensatz hierzu liegt der Fokus in den nachfolgenden Untersuchungen auf der Analyse der Auswirkungen auf die 'realen' Merkmalsvektoren, die unmittelbar zur Spracherkennung verwendet werden. Dazu wird in einem ersten Abschnitt anhand eines rein modellbasierten Klassifikations- und Schätzsystems gezeigt, *dass* sich Sprechgeschwindigkeit sehr gut aus Merkmalsvektoren ablesen lässt, und dies sowohl als Kategorie als auch als kontinuierlicher Wert. Daran anschließend erfolgt in einem zweiten Schritt die Untersuchung, wie sich Sprechgeschwindigkeit in den Erkennermustervektoren niederschlägt und wie diese Information zur Bewertung von Kompensationstechniken eingesetzt werden kann.

3.3.1.1 Systemaufbau

Um den Zusammenhang zwischen spektraler Veränderung und Sprechgeschwindigkeit näher untersuchen zu können, wurde ein Aufbau mit mehreren, parallelen Gauss'schen Mixturmodellen (GMM, s. auch Kap. 5.2.1.2) entwickelt. Der Begriff 'spektral' sei an dieser Stelle als repräsentativ für die Zusammensetzung (linear,cepstral,...) möglicher Merkmalsvektoren verstanden - unabhängig davon, ob der Merkmalsvektor nur rein statische Komponenten umfasst oder durch dynamische Delta(-Delta) Koeffizienten ergänzt wird. Von Interesse ist vor allem die Stärke der Abhängigkeit zwischen Art und Aufbau des Merkmalsvektors und der Sprechrate. Die Entwicklung und Untersuchung dieses Ansatzes wurde auf Ebene der sogenannten Spurts [Mor98] durchgeführt. Abb. 3.3 zeigt den Aufbau des entwickelten Kategoriebestimmungssystems:

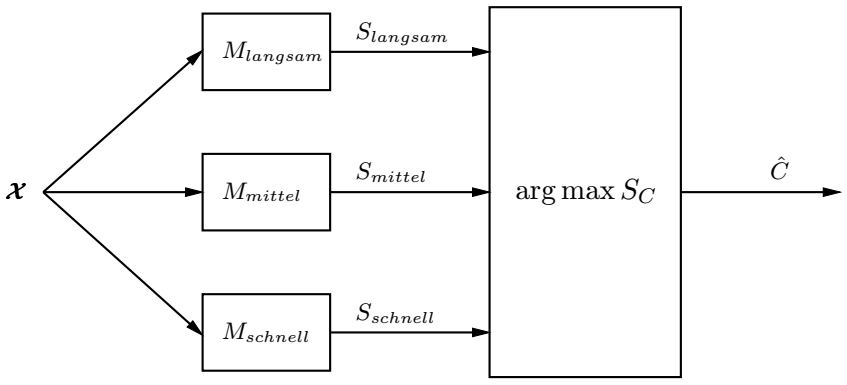

Abb. 3.3: Parallele GMMs mit Maximum-Entscheidung.

Das zugrundeliegende Ziel ist, festzustellen, ob die spektralen Veränderungen konsistent sind und demnach auch konsistent modelliert werden können. So sollte ein Modell, das auf

langsame Sprache trainiert wurde, auch auf neuen, im Training nicht gesehenen Testdaten bei langsamer Sprache bessere Übereinstimmung zeigen, als ein äquivalentes 'schnelles' Modell. Da bei dieser Zielsetzung die Repräsentation der globalen Eigenschaften jeder Kategorie, d.h. ihre allgemeine Lage im Merkmalsraum im Vordergrund steht, wurden bei der stochastischen Modellierung globale, phonemunabhängige GMM-Strukturen herangezogen. Im Gegensatz zu Hidden-Markov-Modellen bleibt bei dieser Repräsentation die zeitliche Abfolge der Merkmalsvektoren unberücksichtigt. Das Entscheidungssystem in Abb. 3.3 beruht im Prinzip auf 3 (respektive 5) parallelen GMMs. Jedes dieser Modelle vertritt eine Sprechgeschwindigkeitskategorie und liefert in der Anwendungsphase eine Bewertung ab, wie gut die aktuelle Äußerung zu dieser Kategorie passt. Ein ähnlicher Ansatz unter Verwendung der Delta-Muster wurde von Martinez in [Mar98] zur Klassifikation auf Satzebene vorgeschlagen. Wie auch in Kapitel 5.2.1.2 erläutert, besteht ein GMM-Modell m aus einer überlagerten Gauss'schen Mixturverteilung.

$$p(\boldsymbol{x}_t|m) = \sum_{k=1}^{K_m} c_{mk}\mathcal{N}(\boldsymbol{x}_t, \boldsymbol{\mu}_{mk}, \boldsymbol{\Sigma}_{mk}) \tag{3.8}$$

Die endgültige Bewertung einer Folge von Vektoren ergibt sich durch die Multiplikation der Frame-weisen Likelihood, d.h.

$$p(\boldsymbol{\mathcal{X}}|m) = \prod_{t=1}^{T} p(\boldsymbol{x}_t|m) \tag{3.9}$$

Durch Logarithmierung erhält man den Score für diese Vektorsequenz als Summe der Framescores:

$$S_m(\boldsymbol{\mathcal{X}}) = \log p(\boldsymbol{\mathcal{X}}|m) = \sum_{t=1}^{T} \log p(\boldsymbol{x}_t|m) \tag{3.10}$$

Der negierte Score $-S_m(\boldsymbol{\mathcal{X}})$ stellt ein Quasi-Distanzmaß dar, und spiegelt den Abstand der zu klassifizierenden Vektoren zum jeweiligen Kategoriemodell wieder. Steht die Bestimmung der Sprechratenkategorie im Vordergrund, kann durch eine implizite Maximum-Entscheidung ein Votum für eine Klasse getroffen werden:

$$\hat{C} = \arg\max_{C} S_C \quad \text{wobei} \quad C \in \{langsam, mittel, schnell\} \tag{3.11}$$

Für das Training der GMM-Modelle kann aufgrund der ausreichenden Menge an Trainingsdaten der ML-Trainingsalgorithmus verwendet werden. Dies ist desweiteren dadurch gerechtfertigt, da bei der Zielsetzung die globale Verteilung zu erfassen, tendenziell nur wenige Gauss'sche Prototypen in einem GMM verwandt werden. Bei 64 Prototypen mit jeweils 2*42 Komponenten bewegt sich die Zahl der freien Parameter bei ca. 5000 je GMM. Zum Vergleich: bei einem Spracherkennungssystem bewegt sich die Gesamtzahl der freien Parameter, die aus dem selben Datenmaterial geschätzt werden müssten, häufig jenseits 10^6. Die robuste Schätzung der GMM-Parameter ist daher unproblematisch.

Ein kritischerer Aspekt ist hier jedoch die Einteilung der Trainingsdaten: Analog zu dem Vorgehen in [Pfa00b] wurde hier ebenfalls eine Unterteilung der Trainingssätze in sogenannte

'Spurts' vorgenommen, wobei hier jedoch die Phonemrate ROS_{Ph} als Sprechgeschwindigkeitsmaß favorisiert wurde. Aus Gründen der Einfachheit sei im folgenden unter dem Begriff 'Spurtrate' v_{Spurt} die Phonemrate ROS_{Ph}^{Spurt} eines Spurts aufzufassen.

$$v_{Spurt} = ROS_{Ph}^{Spurt} = \frac{N_{Ph}}{T_{Spurt}} * 100 \quad \text{in} \quad [Phoneme/s] \tag{3.12}$$

In die GMM-Klassifikation gehen nahezu alle betrachteten Frames ein. Ausnahmen bilden lediglich die Frames, die von einem vorgeschalteten statistischen Sprach-Pause Detektor [Beh95b] als Pause oder als 'unsicher' markiert werden. Der selbe Detektor wird auch zur Einstufung der Merkmalsvektoren des Trainingskorpus herangezogen. Eine Verteilung der Sprechgeschwindigkeiten innerhalb der Trainingsdaten kann Abb. 3.4 entnommen werden.

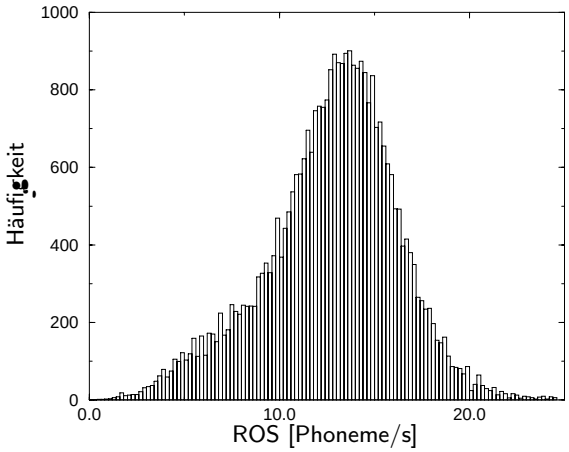

Abb. 3.4: Verteilung der Sprechgeschwindigkeiten (je Spurt) innerhalb der Trainingsdaten.

Abb. 3.4 zeigt eine annähernde Gaussverteilung der Sprechraten mit Mittelwert $\mu_{v_{Spurt}} = 12,765[Phoneme/s]$ und Standardabweichung $\sigma_{v_{Spurt}} = 3,4924[Phoneme/s]$. Eine Unterteilung kann anhand dieser Werte erfolgen:

$$C_{Spurt} = \begin{cases} schnell & \text{wenn } v_{Spurt} > \mu_{v_{Spurt}} + \Delta \\ langsam & \text{wenn } v_{Spurt} < \mu_{v_{Spurt}} - \Delta \\ mittel & \text{sonst} \end{cases} \tag{3.13}$$

wobei $\Delta = \sigma_{v_{Spurt}}$. Eine Einteilung anhand dieser Grenzen führt dazu, dass nur jeweils ca. 15% der Daten den Kategorien 'schnell' bzw. 'langsam' zugeordnet werden. Im Gegensatz zu klassenweisen HMM-Modellen zur Spracherkennung, bei denen der bereits geringe Anteil noch weiter auf einzelne Phonemmodelle bzw. deren Zustände aufgeteilt wird, stehen hier die ganzen Daten für ein Modell mit relativ wenigen Verteilungen zur Verfügung. Die robuste Schätzung der Parameter ist daher unkritisch.

3.3.1.2 Ergebnisse

Zur Generierung geeigneter Referenzdaten wurden die Evaluierungssätze (Eval96) des Verbmobil Korpus unter Verwendung der korrekten Transliteration segmentiert. Basierend auf

dieser Segmentierung konnten die Äußerungen in Spurts eingeteilt werden und die jeweilige Spurtrate bestimmt werden. Als minimale Pausenlänge zur Separierung in Spurts wurden $250ms = 25$ Frames angenommen. Für einen direkten Vergleich wurde die dem Spurt entsprechende Framesequenz als Eingangsdaten für den Klassifikator verwendet. Insgesamt standen auf diese Art und Weise 755 einzelne Vergleichsbewertungen zur Verfügung. Tabelle 3.1 zeigt einen Vergleich bei Verwendung der drei Geschwindigkeitsklassen: langsam, mittel, schnell. Die eingesetzten GMMs haben jeweils 16 Normalverteilungen mit diagonaler Kovarianzmatrix.

Vorverarb.	korrekt [%]	ρ
MFCC42	60.9	0.47
MFCC12	37.8	0.17
MUC66	35.5	0.43

Tab. 3.1: Klassifikationsergebnis bei 3 Geschwindigkeitskategorien und verschiedenen Vorverarbeitungen.

Der den Ergebnissen aus Tabelle 3.1 zugrundeliegende Systemaufbau orientierte sich strikt an dem in Abb. 3.3 dargestellten, d.h. die Entscheidung wird anhand des Scoremaximums getroffen. Wie die Tabelle zeigt, konnten bei der Verwendung der als MFCC42 bezeichneten Vorverarbeitung (s. Abschnitt 1.4) 60.9% aller Spurts richtig zugeordnet werden. Da jedoch auch die Referenz mit einer gewissen Unsicherheit verbunden ist, ist die zweite Spalte der Tabelle aufschlussreicher: Sie gibt den Korrelationskoeffizienten zwischen Referenz und Hypothese an. Der Korrelationskoeffizient lässt sich aus der Verwechslungsmatrix (s. Tab 3.2) bestimmen.

	Hypothese		
Referenz:	langsam	mittel	schnell
langsam	75	8	2
mittel	196	373	69
schnell	8	42	59

Tab. 3.2: Verwechslungsmatrix bei MFCC42-Vorverarbeitung: geschätzte Kategorie gegen Referenzkategorie.

Die Verwechslungsmatrix gibt für jede Kategorie der Referenz die Anzahl der zugeordneten Hypothesen an - ebenso nach Kategorie aufgeschlüsselt. Auffallend ist, dass sehr wenige direkte Verwechslungen zwischen den extremen Sprechgeschwindigkeiten 'schnell' und 'langsam' auftreten. Dies macht deutlich, dass diese Extreme unterschieden werden können. Andererseits treten zwischen den benachbarten Kategorien 'mittel' und 'schnell' bzw. 'mittel' und 'langsam' sehr viele direkte Verwechslungen auf. Dies wird jedoch verständlich, wenn man die Einteilung der Trainingsdaten berücksichtigt. Die Sprechraten sind kontinuierlich verteilt (s. Abb. 3.4), werden aber durch die Festlegung einer Grenze fix auf bestimmte Kategorien quan-

tisiert. Speziell an den Kategoriegrenzen führt dies zu Verwechslungen. Aus der Verwechslungsmatrix in Tab. 3.2 ergibt sich ein Korrelationskoeffizient $\rho = 0.47$ (MFCC42, GMM: 16 NV). Dieser Wert liegt in einem der Literatur vergleichbaren Niveau [Mor97, Mor98].

Ein weiterer bemerkenswerter Aspekt aus Tabelle 3.1 ist der starke Unterschied - speziell des Korrelationskoeffizienten beim direkten Vergleich der Vorverarbeitungen MFCC12 und MFCC42. Bei ersterer werden lediglich die eigentlichen 12 mel-cepstralen Koeffizienten berücksichtigt, wohingegen bei der 42-dimensionalen Vorverarbeitung neben der Nulldurchgangsrate insbesondere noch die Delta- bzw. Delta-Delta Koeffizienten enthalten sind. Ein Wert von $\rho = 0.17$ zeigt, dass es bereits auf der 'spektralen' Ebene zu einer messbaren, primär durch die veränderte Sprechgeschwindigkeit verursachten Abweichung kommt. Eine weitaus stärkere Abweichung der Merkmalsvektoren ergibt sich jedoch, wenn diese um die Ableitungs- sowie Beschleunigungskoeffizienten erweitert werden. Dies lässt deutlich werden, dass sich ein Großteil der Information über die Sprechgeschwindigkeit in der Abweichung der Spektren zueinander niederschlägt. Dieses Ergebnis stützt im Prinzip die Ansätze einer variablen Delta-Berechnung ('Sprechratennormierung') [Pfa00b, Tsu00, Ned01].

Aufgrund der geringen Parameterzahl der verwendeten GMM-Modelle ist die Menge des verfügbaren Trainingsmaterial unkritisch. Es konnte hier auch die Einteilung in mehr als drei Kategorien untersucht werden. Bei fünf Kategorien gilt entsprechend:

$$C_{Spurt} = \begin{cases} \textit{sehr schnell} & \text{wenn } v_{Spurt} > \mu_{v_{Spurt}} + 2\Delta \\ \textit{schnell} & \text{wenn } \mu_{v_{Spurt}} + \Delta < v_{Spurt} \leq \mu_{v_{Spurt}} + 2\Delta \\ \textit{sehr langsam} & \text{wenn } v_{Spurt} < \mu_{v_{Spurt}} - 2\Delta \\ \textit{langsam} & \text{wenn } \mu_{v_{Spurt}} - 2\Delta \leq v_{Spurt} < \mu_{v_{Spurt}} - \Delta \\ \textit{mittel} & \text{sonst} \end{cases} \tag{3.14}$$

Bei einer Gaussverteilung liegen oberhalb bzw. unterhalb von 2σ nicht mehr ausreichend Daten - aus diesem Grund muss Δ hier kleiner gewählt werden. Entsprechend der gemessenen Verteilung wurde ein Wert $\Delta = 2.0[Phone/s]$ angenommen. Tabelle 3.3 zeigt die ermittelten Ergebnisse.

Vorverarb.:	korrekt [%]	ρ
MFCC42	35.3	0.58
MFCC12	23.4	0.23
MUC66	24.4	0.54

Tab. 3.3: Korrekte Zuordnung und Korrelationskoeffizient bei Verwendung von 5 Kategorien.

Bei allen Vorverarbeitungen zeigt sich eine Verschlechterung der reinen Klassifikationsleistung - gemessen an der Referenzkategorie. Aufgrund der höheren Klassenzahl, sowie der Überschneidung an den Klassengrenzen, war dies zu erwarten. Im Gegensatz dazu ergibt sich jedoch eine deutlich höhere Korrelation zwischen hypothetisierter Kategorie und der Referenz.

Wie auch bei der 3-Klassen Einteilung belegen auch diese Experimente den primären sprechgeschwindigkeitsspezifischen Informationsgehalt der Ableitungskoeffizienten: Werden nur die reinen 12 Mel-Cepstren als Merkmalsvektor verwendet, so sinkt der Korrelationskoeffizient auf 0.23 - was aber immer noch auf eine Abhängigkeit hindeutet. Um die Ergebnisse weiter aufzuschlüsseln, kann die Verwechslungsmatrix für 5 Kategorien in Tab. 3.4 betrachtet werden.

	Hypothese				
Referenz:	sehr langsam	langsam	mittel	schnell	sehr schnell
sehr langsam	58	8	5	1	1
langsam	27	52	3	4	0
mittel	37	207	124	25	36
schnell	3	26	59	22	53
sehr schnell	2	6	23	12	38

Tab. 3.4: Verwechslungsmatrix: Zahl der zugeordneten Spurts, aufgeschlüsselt nach Kategorien (MFCC42, 16 NV).

Aus Tabelle 3.4 wird wiederum deutlich, dass zwar zwischen benachbarten Kategorien Verwechslungen auftreten, jedoch sehr wenige zwischen den entgegengesetzten Klassen (sehr)langsam und (sehr)schnell. Dies ist besonders beim Einsatz als Klassifikator zur Selektion der einzusetzenden Erkennermodelle von Bedeutung, da hier die Wahl der entgegengesetzten Modelle stark negative Auswirkungen auf die Erkennungsleistung hat.

Der Klassifikatoraufbau ist dazu geeignet, zu untersuchen, welche Phoneme besonders durch die Sprechgeschwindigkeit beeinflusst sind. Dazu wurde anhand einer Segmentierung der Trainings- bzw. Testsätze für jedes Phonem individuell die Stärke der Score-Abweichung bestimmt [Fal00b]. Die Score-Abweichung ergibt sich als der mittlere Score-Unterschied aus der Bewertung mit dem richtigem Modell und jeweilig 'entgegengesetztem' (schnell $\leftrightarrow$ langsam) Modell. Dieses Maß drückt aus, wie stark das entgegengesetzte Modell vom richtigen abweicht.

Referenz:	Trainingssätze
langsam	/m/, /n/, /e:/, /E:/, /2:/, /i:/, /a:/
schnell	/U/, /aU/, /b/, /d/, /o:/, /z/
	Testsätze
langsam	/m/, /n/, /e:/, /E:/, /S/, /i:/
schnell	/z/, /b/, /d/, /9/, /I/, /O/, /a/, /U/

Tab. 3.5: Phoneme mit den höchsten Score-Abweichungen.

Tabelle 3.5 zeigt diejenigen Phoneme, für die sich die stärksten Abweichungen ergeben - jeweils aufgeschlüsselt nach schneller bzw. langsamer Referenzkategorie. Deutlich wird, dass

bei eher langsamer Sprache insbesondere die Nasale sowie die Vokale /E:/ /2:/ /i:/ /a:/ betroffen sind, wohingegen bei schneller Sprache eher Vokale wie /U/ und /o:/ beeinflusst werden. Auffallend ist, dass bei letzterer Kategorie besonders Plosive auftreten.

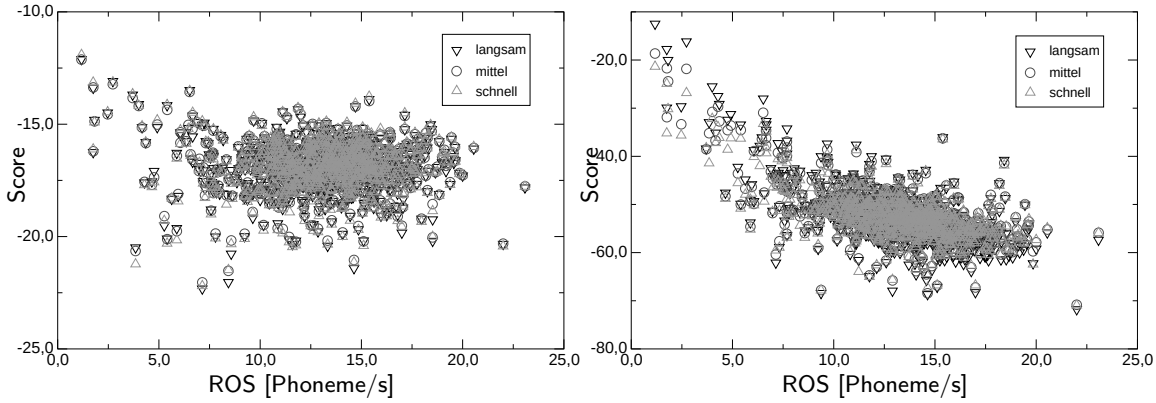

Abb. 3.5: Vergleich der Modellscores bei MFCC12- (links) und MFCC42-Merkmalsvektoren, abhängig von der Sprechgeschwindigkeit.

Ein weiterer Aspekt kann den Streuplots in Abbildung 3.5 entnommen werden. Die Abbildungen zeigen die Modellscores auf Basis zweier unterschiedlich aufgebauter Merkmalsvektoren. Links werden ausschließlich die 12 statischen Mel-cepstralen Komponenten (MFCC12) verwendet. Im rechten Streuplot sind darüber hinaus auch dynamische Komponenten im Merkmalsvektor integriert: 12 Delta-, 12 DeltaDelta-Koeffizienten, NDR und die Gesamtenergie, sowie deren 1. und 2. Ableitung (MFCC42). Dargestellt sind die Scores der Modelle 'schnell, mittel und langsam', ausgewertet jeweils auf allen Spurts. Für einen Spurt mit einer anhand der Segmentierung ermittelten Phonemrate ist immer eine Dreier-Gruppe maßgeblich: Das nach unten zeigende Dreieck (∇) symbolisiert das langsame Modell, das noch oben zeigende ($\triangle$) entsprechend das schnelle Modell. Der Kreis steht für das mittlere Modell.

Entsprechend des zugrundeliegenden Ansatzes sollten tendenziell bei langsamer Sprache eher die langsamen' GMMs (∇) die höchsten Scores aufweisen. Mit zunehmender Sprechgeschwindigkeit sollten dann die 'mittleren' ($\circ$) und schließlich die auf schneller Sprache trainierten GMMs (Δ) dominieren. Bei der MFCC42-Vorverarbeitung (rechts) ist dies sehr gut erfüllt. Bei der linken, ausschließlich statischen Vorverarbeitung ist dies offensichtlich nur bedingt gegeben.

Ein weiterer Punkt wird beim direkten Vergleich der obigen Bilder offenkundig: Sobald die dynamischen Merkmale integriert werden, ergibt sich mit steigender Sprechgeschwindigkeit ein starker Abfall der Scores, der für alle Modelle, unabhängig von der Kategorie, zu beobachten ist. Bei der Verwendung rein statischer Features ist dieser Effekt bei weitem weniger stark ausgeprägt. Dieser Abfall kann auch durch die angepasste Modellierung ('schnelle' Modelle) nicht kompensiert werden. Da die an dieser Stelle eingesetzten GMM-Modelle im Prinzip ebenfalls eine Modellierung des Merkmalsraums vornehmen, kann dieses Ergebnis auch auf die Generierung von HMM-Modellen zur Spracherkennung übertragen werden.

Auch bei diesen Modellen sollte, bei Verwendung von dynamischen Merkmalen, ein Absinken der Modellierungsgenauigkeit bei schneller Sprache miteinhergehen. Eine genauere Analyse dieses Effekts folgt in Abschnitt 3.4.1.

3.3.2 Kontinuierliche Sprechgeschwindigkeit

3.3.2.1 Systemaufbau

Die Experimente mit 5 oder mehr Kategorien haben gezeigt, dass bei einer feineren Unterteilung der Klassen eine höhere Korrelation erreichbar ist. Dieses Ergebnis wirft die Frage auf, ob nicht ein Rückschluss auf den eigentlichen Wert der Sprechgeschwindigkeit, z.B. in [Phoneme/s] möglich ist. Um diese Frage näher zu untersuchen, wurde der Systemaufbau in Abb. 3.6 herangezogen:

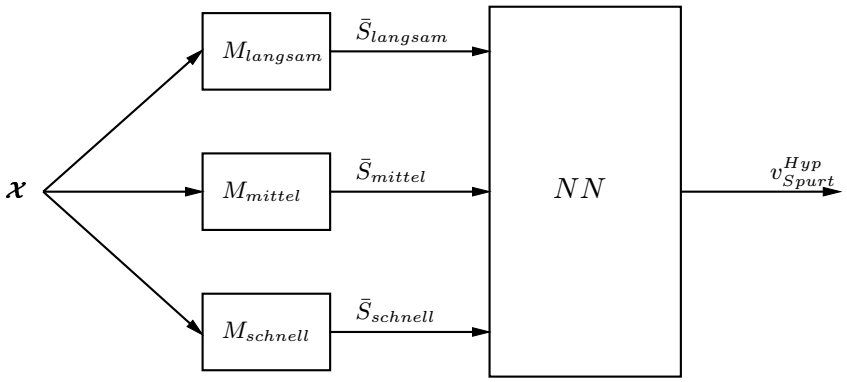

Abb. 3.6: Parallele GMMs mit Abbildungsfunktion, realisiert durch ein neuronales Netz (NN).

Analog zum Systemaufbau zur Kategoriebestimmung wird ebenfalls eine Eingangsschicht aus 3 (bzw. 5) parallelen GMMs verwendet. Diese liefern ein Ähnlichkeitsmaß der aktuellen Äußerung zur jeweiligen Kategorie, da jeder GMM-Ausgang mit seinem Ausgangsscore ein *kontinuierliches* Maß erzeugt. Die Scoreberechnung kann als eine Abbildung aufgefasst werden, d.h.:

$$R^1 \longmapsto R^1 : \qquad \bar{S}_m = f_m(v_{Spurt}) \tag{3.15}$$

Da jedes der m parallelen GMMs eine Bewertung abgibt, kann dies als eine Abbildung der Sprechgeschwindigkeit in einen 3-dimensionalen Raum angesehen werden.

$$R^1 \longmapsto R^3 : \qquad \boldsymbol{f}(v_{Spurt}) = \boldsymbol{f}_{map}(v_{Spurt}) = \begin{pmatrix} f_{langsam}(v_{Spurt}) \\ f_{mittel}(v_{Spurt}) \\ f_{schnell}(v_{Spurt}) \end{pmatrix} \tag{3.16}$$

Die *unbekannte* (im Anwendungsfall) Sprechgeschwindigkeit v_{Spurt} eines jeden Sprachabschnitts beschreibt also einen Punkt in diesem mehrdimensionalen Raum. Gesucht ist eine

inverse Abbildung f_{map}^{-1}, die diesen Punkt wieder auf ein eindimensionales, kontinuierliches Maß zurückführt:

$$R^3 \longmapsto R^1 : \qquad v_{Spurt}^{Hyp} = f_{map}^{-1}(\boldsymbol{f}(v_{Spurt})) = f_{map}^{-1}(\boldsymbol{f}_{map}(v_{Spurt})) \tag{3.17}$$

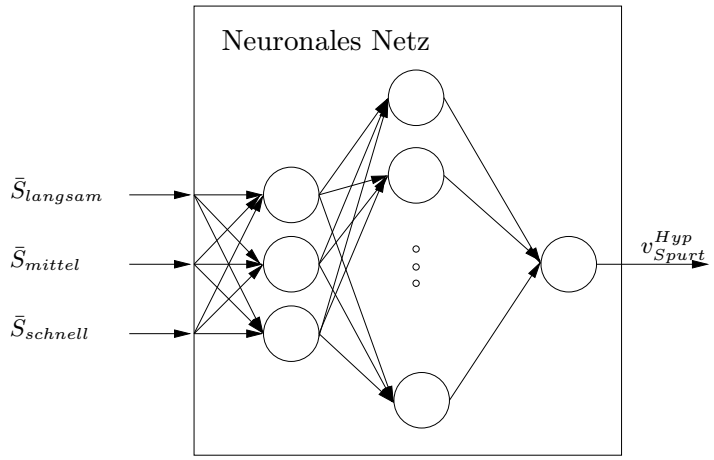

Abb. 3.7: Abbildungsfunktion f_{map}^{-1} realisiert durch ein Neuronales Netz.

Realisiert wurde diese Abbildungsfunktion mittels eines neuronalen Netzes (NN). In Abb. 3.7 ist dieses Netz schematisch dargestellt. Es verfügt über 3, respektive 5 Eingänge - je nach Anzahl der Sprechgeschwindigkeitskategorien. Intern besteht das Netz aus einer Eingangsschicht mit N_{in}^{NN} Neuronen, einer Ausgangsschicht mit $N_{out}^{NN} = 1$ Neuron, sowie einer optionalen versteckten Schicht mit N_{hidden}^{NN} Neuronen. Das Netzwerk ist in dieser Anwendung nicht als Klassifikator ausgelegt, sondern soll die Funktion f_{map}^{-1} in Gleichung 3.17 möglichst genau approximieren. Als Trainingsverfahren für das Netzwerk wurde ein Backpropagation-Algorithmus implementiert [ReW96]. Zu diesem Zweck wurde die folgende quadratische Fehlerfunktion definiert:

$$e = (v_{Spurt}^{Ref} - v_{Spurt}^{Hyp})^2 \tag{3.18}$$

Als Referenzgeschwindigkeit v_{Spurt}^{Ref} wurde die anhand einer Segmentierung ermittelte Phonemrate eines Spurts angesetzt (Gl. 3.12). Die geschätzte Spurtrate v_{Spurt}^{Hyp} ergibt sich am Ausgang des NN, das die Modellscores auf den hypothetisierten Wert v_{Spurt}^{Hyp} abbildet:

$$v_{Spurt}^{Hyp} = f_{map}^{-1}(\bar{S}_{langsam}, \bar{S}_{mittel}, \bar{S}_{schnell}) \tag{3.19}$$

Als Eingangswerte des NN müssen die längennormalisierten Scores $\bar{S}_m$ eingesetzt werden, um die möglicherweise unterschiedlichen zeitlichen Längen der Spurts auszugleichen.

$$\bar{S}_m = \frac{1}{T_{Spurt}} S_m = \frac{1}{T_{Spurt}} \sum_{t=1}^{T_{Spurt}} \log p(\boldsymbol{x}_t|m) \tag{3.20}$$

Beim Backpropagation-Algorithmus handelt es sich im Prinzip um ein Gradientenabstiegsverfahren: Der Eingangsvektor wird durch das kontinuierliche, d.h. differenzierbare

Netzwerk auf den Ausgangswert v_{Spurt}^{Hyp} abgebildet. Aus dem resultierenden Fehler e, mit dem der Ausgangswert vom Soll(=Target)wert v_{Spurt}^{Ref} abweicht, kann der Gradient ∇ bestimmt werden, in dessen negativer Richtung die Parameter verschoben werden müssen, um den Fehler zu verringern.

$$\lambda_i^{NN}(k+1) = \lambda_i^{NN}(k) - \epsilon\nabla\lambda_i^{NN}(k) - \tau\nabla\lambda_i^{NN}(k-1) \tag{3.21}$$

Das Training des Netzwerks kann sequentiell oder im sogenannten Batchmodus erfolgen, wobei sich ein sequentieller Ansatz als günstiger erwiesen hat. In beiden Fällen wird dieses Vorgehen solange iteriert, bis der Algorithmus konvergiert. Der Prozess wird als konvergent angenommen, wenn die Änderung des Gesamtfehlers $\sum e_j$ von einem Durchlauf aller Muster auf den nächsten unter eine gegebene Schwelle e_{min} fällt. Um die Wahrscheinlichkeit zu verringern, dass die Netzparameter gegen ein lokales Minimum konvergieren, hat es sich als vorteilhaft erwiesen, den Gradientenabstieg in Gl. 3.21 um einen sogenannten Momentum-Term zu erweitern. Der mit τ gewichtete Momentum-Term gibt - durch die Berücksichtigung der Vergangenheit des Gradienten - den Parametern sozusagen eine gewisse “Trägheit”, so dass der Gradientenabstieg sie u.U. über ein lokales Minimum der Fehlerfunktion hinwegtragen kann.

Für die Eingangsneuronen wurde eine nichtlineare Aktivierungsfunktion gewählt. Infrage kommen hierfür insbesondere eine sigmoide oder eine hyperbolische Aktivierung. Um eine sinnvolle und v.a. schnelle Konvergenz zu erhalten, müssen die statischen Parameter der Eingangsschicht sorgfältig gewählt werden. Bei beiden Funktionen ergibt sich nur im Ursprung ein nennenswerter Beitrag des Gradienten. Die Eingangsscorewerte bewegen sich im Mittel bei etwa -52 (MFCC42), müssen also in einem ersten Schritt einer Translation unterworfen werden, um sie nahe zum Ursprung zu bringen. Diese Translation kann mittels eines zusätzlichen Eingangs erreicht werden dessen Wert konstant zu -1 gesetzt wird. Das Gewicht w_0 dieses Eingangs kann innerhalb des Trainings mit gelernt werden.

Ein zweiter Parameter, der Einfluss auf die Konvergenz hat, ist die ’Breite’ α der Aktivierungsfunktion. Mit ihr muss der Streubereich der Eingangsvektoren abgedeckt sein, damit jeder Vektor noch einen Beitrag zum Gradienten liefern kann.

3.3.2.2 Ergebnisse

Die Experimente zeigten, dass ein Netzwerk mit ca. 20 Neuronen in der Eingangsschicht, ca. 3-4 Neuronen in der Zwischenschicht, sowie dem einzelnen Neuron (vgl. Abb. 3.7) der Ausgangsschicht gut geeignet ist. Die Verwendung eines versteckten Layers hat jedoch starke Auswirkungen auf die Rechendauer des Trainingsprozesses. Da die Gradienten einer Schicht von denen der Nachfolgerschicht abhängen, werden diese mit steigender Zahl von Schichten sehr klein - speziell wenn sich der Fehler einem Minimum nähert. Die Konvergenzgeschwindigkeit sinkt damit stark ab. Versuche zeigten, dass ein NN mit nur einer Eingangsschicht mit ca. 50 Neuronen und keiner Zwischenschicht ähnlich gute Ergebnisse zeigt - verbunden jedoch mit einer deutlich schnelleren Konvergenz.

Abb. 3.8 zeigt einen Streuplot zwischen der Referenzrate gemäß der Segmentierung und der geschätzten Phonemrate. Die Auswertung erfolgte mit den Eval96-Testdaten.

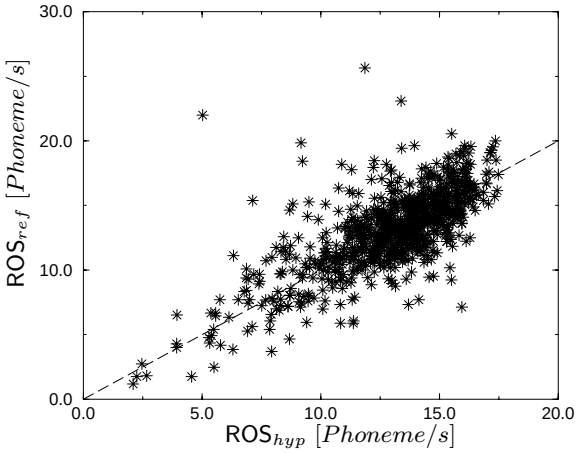

Abb. 3.8: Streuplot zwischen Referenzrate und hypothetisierter Phonemrate (Eval96-Testdaten).

Abhängig von der Anzahl der Schichten bzw. Neuronen des NN konnte ein Korrelationskoeffizient zwischen 0.66 und 0.70 zwischen Referenz und Hypothese erzielt werden. Dieser Wert zeigt deutlich die starke Abhängigkeit der Mustervektoren von der aktuellen Sprechrate. Für die GMM-Eingangsschicht wurden drei Modelle mit je 64 NV trainiert.

3.3.3 Erweiterung für gleichzeitige Geschlechtsbestimmung

Wie die in den Kapiteln 2.5 bzw. 5.3.5.1 beschriebenen Experimente mit Eigenvoices zeigen, weist das Sprechergeschlecht einen sehr starken Einfluss auf die akustischen Merkmale auf. Um diesen Einfluss zu berücksichtigen, wurde das System um sprecherspezifische Modelle erweitert. Abb. 3.9 zeigt den schematischen Aufbau.

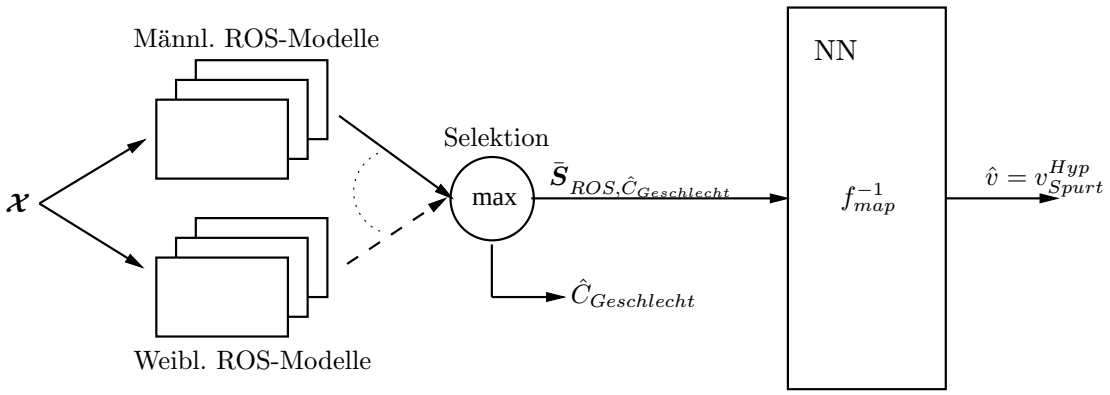

Abb. 3.9: Erweiterter Systemaufbau zur gleichzeitigen Bestimmung der Sprechgeschwindigkeit und des Sprechergeschlechts.

Im dargestellten Aufbau ist jedes der ursprünglich sprecherunabhängigen GMM-Modelle (GI, engl.: “Gender Independent”) durch zwei geschlechtsspezifische Modelle (GD, engl.:

"Gender Dependent") ersetzt. Der Erkennungsvorgang ist um eine zwischengeschaltete Entscheidungseinheit erweitert, da die Abbildungsfunktion nach wie vor die Scores von drei sprechgeschwindigkeitsabhängigen Modellen verarbeiten soll. Dementsprechend werden als Input des NN bzw. der Kategorieentscheidungslogik nur die Scores des "besseren" Geschlechts ausgewählt, d.h.

$$\hat{C}_{Geschlecht} = \arg \max_{C_{Geschlecht}} \bar{S}_{C_{ROS},C_{Geschlecht}} \tag{3.22}$$

$$\bar{\boldsymbol{S}}_{ROS,\hat{C}_{Geschlecht}} = [\bar{S}_{langsam,\hat{C}_{Geschlecht}}, \bar{S}_{mittel,\hat{C}_{Geschlecht}}, \bar{S}_{schnell,\hat{C}_{Geschlecht}}]^T \tag{3.23}$$

In Experimenten konnte verifiziert werden, dass sich mit diesem Systemaufbau der Korrelationskoeffizient gegenüber dem GI-Ansatz um ca. 7% relativ steigern lässt. Tabelle 3.6 schlüsselt die Ergebnisse genauer auf.

GMM-Modelle	GI	GD
ρ	0.70	0.75

Tab. 3.6: Korrelationskoeffizient zwischen geschätzter Phonemrate und Referenzphonemrate bei Verwendung von GI- und GD-GMMs.

Gleichung 3.22, die in erster Linie zur Auswahl der passenden Modellgruppe dient, zeigt, dass bei diesem Ansatz gleichzeitig eine Bestimmung des Sprechergeschlechts mit einhergeht. In einem Erkennungsvorgang können also sowohl die Sprechgeschwindigkeit als auch das Sprechergeschlecht geschätzt werden. Von Interesse ist daher der Vergleich dieses Ansatzes mit einem reinen Geschlechtserkennungsystem. Letzteres wurde mit 2 geschlechtsspezifischen GMM-Modellen realisiert.

GMM-Modelle	ROS-abh.	ROS-unabh.
GFR [%]	6.9	0.9

Tab. 3.7: Vergleich der Geschlechtserkennungsfehlerrate (in [%]) bei sprechgeschwindigkeitsabhängigen bzw. unabhängigen Modellen.

Tabelle 3.7 macht deutlich, dass die sprechgeschwindigkeitsabhängige Modellierung der Geschlechtsmodelle im Gegenzug auch für die Bestimmung des Sprechergeschlechts von Vorteil ist. So ließ sich auf diese Weise die Fehlerrate bei der Geschlechtsbestimmung bzgl. der Eval96-Testdaten signifikant um 6% absolut verringern. Die Geschlechtsfehlerrate $GFR = \frac{N_{falsch}}{N_{gesamt}}$ bezeichnet hier das Verhältnis aus Zahl der falsch klassifizierten Turns (bzw. Sequenzen) zur Gesamtzahl der Turns.

3.4 Untersuchungen zur Lokalen Sprechgeschwindigkeit: lokale Scoremaße

Im Zentrum dieses Abschnitts stehen Untersuchungen, die zum Ziel haben die stochastische, akustische Modellierung mittels HMM zu bewerten. Einige Autoren [Mor97, Mar98, Pfa98b, Pfa00b, Fal99, Wre01] haben bereits festgestellt, dass die Performanz von Spracherkennungssystemen bei schneller Sprache stark leidet. Gemessen wird diese in Form der Wortfehlerrate (WER) oder der Wortakkuratheit (WA, engl. 'word accuracy'). Bei schneller Sprache nimmt die WER - wobei die Angaben variieren - bezogen auf die WER bei durchschnittlicher Sprechrate, um etwa $20-40\%$ relativ zu. Zu der Frage, wie sich die Fehlerrate bei langsamer Sprache verhält, gibt es jedoch unterschiedliche Angaben. Insbesondere Martinez [Mar98] berichtet von einer starken Zunahme der Fehlerrate auf der von ihm gesammelten TRESVEL Datenbasis. Diese wurde speziell für Sprechgeschwindigkeitsuntersuchungen zusammengestellt und enthält Äußerungen in drei Sprechgeschwindigkeitskategorien. Die Sprecher wurden hierbei explizit gebeten langsam, mittel bzw. schnell zu artikulieren. Der Autor berichtet über Sprachartefakte und Artikulationsverhaltensweisen der Sprecher bei langsamer Sprechweise, die den Bereich normaler, zwischenmenschlicher (gelesener oder gesprochener) Aussprache verlassen und bereits in den Bereich der sogenannten 'Hyperartikulation' [Sol98, Sol00] gerechnet werden können.

Laut Martinez tendieren Sprecher - wenn explizit darum gebeten - dazu, langsame Sprache auf zwei prinzipiell unterschiedliche Arten zu artikulieren. Ein Teil der Probanden erhöht die Pausenlänge unter gleichzeitiger, annähernder Beibehaltung der Phonemdauer. Die Sprechweise wirkt in diesem Fall abgehackter. Ein anderer Teil der Sprecher tendiert eher dazu, die Pausenlänge nahezu unverändert zu lassen, im Gegenzug jedoch die Phonemdauern zu strecken - was in stärkerem Maße Auswirkungen auf Vokale hat. Im Gegensatz zu den vorgenannten Autoren zeigen die, in den Untersuchungen von Pfau [Pfa98a] oder Wrede [Wre01] ermittelten Erkennungsergebnisse Verbesserungen - oder zumindest keine nennenswerte Verschlechterung bei langsamer Sprache. Mit der ausschlaggebende Grund für diese Ergebnisse dürfte in der jeweils verwendeten Datenbasis zu sehen sein. Pfau einerseits verwendete mit dem Verbmobil Korpus Daten, die nahezu ausschließlich spontansprachliche Ausdrucksformen enthalten, Wrede andererseits die SLACC-Datenbasis ("Spoken LAnguage Car Control"), die primär gelesene Sprache umfasst. Beide Korpora schließen also vermutlich keine, oder nur in geringem Umfang, Artikulationsformen ein, die als Hyperartikulation aufgefasst werden könnten.

Von einigen Autoren wurden bereits Methoden zur Kompensation des Leistungsabfalls bei schneller Sprache vorgeschlagen. Untersucht wurden beispielweise Ansätze zur Verweildauermodellierung [Mar97] - meist in Form von angepassten HMM-Transitionsgewichten. Die erreichten Verbesserungen fielen allerdings sehr gering aus. Ein anderes Vorgehen, das auch in dieser Arbeit vorgestellt wird, wurde in [Fal99] präsentiert. Der Kerngedanke dieser Arbeit bestand darin, durch eine optimierte, cluster-basierte Initialisierung der Gaussparameter angepasste HMM-Modelle zu erzeugen. Das zugrundeliegende Konzept, Modelle für festgelegte Sprechgeschwindigkeitsklassen zu generieren, stellt das meistdiskutierte Vorgehen zur Anpas-

sung dar. Die Idee hierbei ist, das Trainingsmaterial in vordefinierte Klassen zu teilen und für bzw. mit diesen Daten individuelle HMMs zu trainieren. In der Erkennungsphase könnte dann, anhand der aktuell vorherrschenden Sprechgeschwindigkeit, das geeignete HMM-Set ausgewählt und verwendet werden.

Die vorgestellten Ansätze unterscheiden sich dabei einerseits in der Art, wie die Parameter der jeweiligen Gaussverteilungen geschätzt werden und andererseits im Auffinden passender Modellstrukturen. So untersuchte Pfau [Pfa98b] schwerpunktmäßig, wie sich durch ML- und MAP-Training von Monophonmodellen angepasste HMMs erzeugen ließen. Die erzielten Verbesserungen der WER bei schneller Sprache lagen dabei im Bereich von etwa 2% absolut. Das Training mittels MAP findet sich auch bei Zheng et al. in [Zhe00]. Ein anderer von Pfau vorgeschlagener Weg ist der Einsatz einer Vokaltraktlängennormierung (VTLN, engl.: 'vocal tract length normalization') [LeL96, Pfa00a, Pfa00b] zur Eliminierung von sekundären, sprecherspezifischen Einflüssen auf die Sprechgeschwindigkeit.

Die meisten der Autoren beschränken sich bei der Klassenfestlegung auf die drei Kategorien langsam, mittel und schnell. Der Hintergrund hierfür ist einerseits in der, durch die Teilung der Trainingsdaten bedingten, Abnahme der Datenmenge je Klasse zu sehen, die eine robuste Parameterestimierung erschwert. Der zweite maßgebliche Grund liegt in der robusten Schätzung der Sprechgeschwindigkeit. Dies gilt sowohl für die Trainingsphase, bei der die Einteilung der Trainingsdaten erfolgt, als auch für die Erkennungsphase, in der die Selektion der passenden Klassenmodelle erfolgen muss. Eine Ausnahme bildet in diesem Zusammenhang die Studie von Wrede [Wre01], in der die Autorin die Abhängigkeit der WER von der Modellselektion während der Erkennungsphase untersucht. Sie berücksichtigt auch den Fall von mehr als drei Kategorien. Konkret werden von ihr bis zu 18 Gruppen erzeugt und trainiert. Hierzu muss angemerkt werden, dass die Autorin ein Spracherkennungssystem, basierend auf semikontinuierlichen HMM-Modellen, mit 512 Gaussverteilungen einsetzt, welches aufgrund des starken Tyings auf eine Teilung der Trainingsdaten tendenziell unempfindlich(er) reagiert. Die Untersuchung umfasst neben verschiedenen (globalen) Sprechgeschwindigkeitsmaßen auch die implizite Auswahl anhand des besten Scores. Die Autorin kommt zu dem Schluss, dass verglichen mit dem impliziten Ansatz, keines der expliziten Auswahlkriterien eine bessere Performanz zeigt.

Im Gegensatz zu reinen Monophonmodellen, erlaubt die Generierung von Triphonmodellen unterschiedliche Ansatzpunkte zur Erzeugung einer klassenweisen Modellierung. Eine Vorstellung und Diskussion von Verfahren zur Zustandsgruppierung, die speziell das sogenannte Zustandstying mittels Entscheidungsbäumen zum Schwerpunkt haben, findet sich in dieser Arbeit in Kapitel 4. Ein Ansatz, der Sprechgeschwindigkeit als mögliches Entscheidungskriterium im Entscheidungsbaumprozess integriert, wurde in [Fal00c] vorgestellt und wird ebenfalls in dem genannten Kapitel diskutiert.

Ein generelles "Problem" bei der Beurteilung von Auswirkungen der Sprechgeschwindigkeit bzw. der Wirksamkeit von Kompensationsmaßnahmen ist die Bewertung anhand der Wortfehlerrate. Letztendlich ist dieses Maß zwar das Kriterium, anhand dessen sich alle

Erkennungssysteme bewerten lassen müssen - aber zur Analyse und Beurteilung wo und wie innerhalb des Erkennungssystems Kompensationstechniken greifen, bzw. sich durch die Sprechgeschwindigkeit verursachte Veränderungen auswirken, ist die Angabe der WER nur unzureichend geeignet. Ein Spracherkennungssystem kann als ein komplexes System mit untereinander abhängigen Komponenten gesehen werden. Dies beginnt mit der Vorverarbeitung und der Merkmalsbildung, bei der sich Mel-Cepstren einschließlich 1. und 2. Ableitung, zur Erfassung der zeitlichen Dynamik, als Merkmale durchgesetzt haben. Bei der akustischen Modellbildung wird versucht, mittels Gauss'scher Verteilungen, die WDF der Laute im (hier mel-cepstralen) Merkmalsraum zu repräsentieren. Phonetische sowie zeitliche (Dauer)Änderungen, die durch die Sprechgeschwindigkeit verursacht sein können, haben dadurch direkten Einfluss auf die Lage der Merkmalsvektoren und damit auf die Übereinstimmung mit der parametrisierten Repräsentation. Dies konnte bereits in den vorangegangenen Abschnitten dieses Kapitels gezeigt werden. Die Angabe der Veränderung der WER lässt hier nur ungenügende Aussagen über die Art der Veränderung bzw. deren Abhängigkeit von den Systembestandteilen zu. Dies resultiert teilweise daraus, dass die statistische Darstellung im Erkennungssystem durch weitere Wissenskomponenten wie Lexikon und Sprachmodell ergänzt wird. Gewisse Auswirkungen bei steigender Sprechgeschwindigkeit, wie beispielsweise Elisionen oder Koartikulation, werden u.U. auch (bzw. erst) durch diese Komponenten wirksam. Zusammenfassend ausgedrückt, erlaubt die Angabe der WER keinen näheren Aufschluss über den "Einflussort" innerhalb des Spracherkennungssystems, der primär zu einer Verschlechterung geführt hat.

Aufgrund dieser mangelnden Aussagekraft sollen in den folgenden Abschnitten einige Maße bzw. Kriterien vorgestellt werden, die eine genauer lokalisierbare Aussage ermöglichen. Der Fokus dieser Arbeit soll auf die akustisch-phonetische Modellierung mittels HMM und Gauss'scher Normalverteilungen gelegt werden. Der Einfluss auf die lexikalische Modellbildung (z.B. Aussprachevarianten) und das Sprachmodell wird daher ausgeblendet.

Vorgestellt wurde mit der lokalen Sprechgeschwindigkeit in Abschnitt 3.2.4 ein Kriterium, das eine genauere Erfassung und Lokalisierung der zeitlichen Veränderung der Sprechgeschwindigkeit erlaubt. Dies ist insofern von Bedeutung, als dadurch die Ursache der Auswirkungen auf die Modellierung auch zeitlich eindeutiger zugeordnet werden kann.

Als Kriterien zur Bewertung der Auswirkung von schneller Sprache auf die statistische Modellbildung sollen im folgenden der akustische Score (als Logarithmus der Emissionswahrscheinlichkeitsdichte) bzw. dessen mittelfristige Schwankung ("lokaler Score") vorgestellt werden. Während der Score noch stark von lokalen Grenzveränderungen zwischen Einzellauten bestimmt wird, kann durch eine Tiefpassfilterung die mittelfristige Änderung der Modellierungsgenauigkeit erfasst werden. Die Filterbandbreite kann dabei an der möglichen Änderungsgeschwindigkeit der Sprechgeschwindigkeit ausgerichtet werden, um deren Dynamik erfassen zu können. Da die Scoremaße im Prinzip nur die Auswirkungen auf das 'korrekte' Modell beschreiben und konkurrierende Modelle vernachlässigen, werden sie durch zwei Konfidenzmaße ergänzt, welche die Beschreibung der Unterscheidbarkeit (Diskriminanz) der Modelle zum Inhalt haben.

3.4.1 Lokaler Score Mittelwert (LAS)

In Anlehnung an die Festlegung und Bezeichnung der lokalen Sprechgeschwindigkeit (LSR, vgl. Abschnitt 3.2.4) lässt sich für den Score ein lokales Maß festlegen. Im folgenden sei unter dem lokalen Score die mittelfristige Schwankung der Frame-weisen Scorewerte verstanden. Interpretiert man die einzelnen Scorewerte als Abtastwerte, also als diskrete Folge, so lässt sich diese Schwankung anhand einer Tiefpassfilterung des 'Scoresignals' bestimmen. Die Eingangsfolge weist eine Abtastfrequenz von $f_A = \frac{1}{\Delta t} = \frac{1}{10ms} = 100Hz$ auf. Analog zur Berechnung der LSR (vgl. Abb. 3.1) wird die Filterung mittels eines "Running-Average"-Rechteckfensters der Breite N_F realisiert. Am Ausgang ergibt sich der lokale Scoremittelwert LAS (engl. 'Local Average Score') somit zu:

$$LAS(n) = \frac{1}{N_F} \sum_{j=n-\frac{N_F}{2}}^{n+\frac{N_F}{2}-1} S_{\hat{m},\hat{s}}(j) = \frac{1}{N_F} \sum_{j=n-\frac{N_F}{2}}^{n+\frac{N_F}{2}-1} \log p(\boldsymbol{x}_j|\hat{m},\hat{s}) \tag{3.24}$$

$\hat{m}$ bezeichnet das korrekte Modell und $\hat{s}$ den zugehörigen HMM-Zustand zum Zeitpunkt j. Beide können durch eine Forced-Viterbi Segmentierung der Äußerung ermittelt werden. Das Fenster ist bezüglich des Berechnungszeitpunkts n symmetrisch angeordnet und erfasst N_F Frames. Abb. 3.10 zeigt beispielhaft den Verlauf des LAS für Turn 'g091a000'. Die Übertragungsfunktion des Rechteckfensters stellt einen Tiefpass dar, dessen Bandbreite in etwa 1Hz beträgt. Es werden also nur Veränderungen erfasst, die in ihrer Ausdehnung größenordnungsmäßig über einer Sekunde liegen.

Aufgrund der dadurch drastisch reduzierten Bandbreite könnte die Abtastfrequenz des LAS 10..20:1 durch Downsampling abgesenkt werden. Dies entspricht einem Fenstervorschub von 10..20 Frames.

3.4.2 Lokale Konfidenz

Die Definition des LAS ist primär auf die Auswirkungen auf die Likelihoodschätzung der (korrekten) Modelle ausgerichtet. Es kann argumentiert werden, dass die Angabe der akustischen Likelihood allein noch nicht aussagekräftig genug ist. Konkurrierende Modelle könnten ja der selben Veränderung unterworfen sein, wodurch der 'Abstand' zum korrekten Modell prinzipiell erhalten bliebe. Die Einfluss der Sprechgeschwindigkeit wäre in diesem Fall faktisch ohne Auswirkung auf die Erkennung. Die Verschlechterung der Worterkennungsraten für schnelle Sprache deutet allerdings bereits an, dass dies nur bedingt zutreffen kann. Nichtsdestoweniger könnten noch weitere Effekte (z.B. Elisionen,Assimilationen) zu der Verschlechterung führen.

Um zu untersuchen, inwieweit die Entscheidungssicherheit der akustischen Modelle durch die lokale Sprechgeschwindigkeit beeinflusst wird, wurden 2 Konfidenzmaße LC_{T1} und LC_{mean} eingeführt und untersucht.

$$LC_{T1}(n) = \frac{1}{N_F} \sum_{j=n-\frac{N_F}{2}}^{n+\frac{N_F}{2}-1} \Delta S_{T1}(j) \tag{3.25}$$

$$\begin{aligned}\Delta S_{T1}(j) &= S_{\hat{m},\hat{s}}(\boldsymbol{x}_j) - \max_{m\neq\hat{m},s} S_{m,s}(\boldsymbol{x}_j) = \qquad (3.26)\\ &= \log p(\boldsymbol{x}_j|\hat{m},\hat{s}) - \max_{m\neq\hat{m},s} \log p(\boldsymbol{x}_j|m,s) = \\ &= \log \frac{p(\boldsymbol{x}_j|\hat{m},\hat{s})}{\max_{m\neq\hat{m},s} p(\boldsymbol{x}_j|m,s)} \qquad (3.27)\end{aligned}$$

LC_{T1} beschreibt die Scoredistanz des korrekten Modells zum jeweils am stärksten konkurrierenden Modell, welches auch als 'Top-1'-Modell (T1) bezeichnet werden kann. Das korrekte Modell ergibt sich jeweils anhand einer Forced-Viterbi Segmentierung der Referenztranskription. Das T1-Modell wird lokal für jeden Frame individuell bestimmt und ergibt sich nach einer Maximumsuche über alle Zustände aller Konkurrenzmodelle. Da LC_{T1} auch die Distanz zu Modellen enthält, die als phonetisch ähnlich bezeichnet werden können und die oftmals auch als Aussprachevarianten angesehen werden (z.B. /a/ ↔ /a:/), wurde LC_{mean} als zweites Konfidenzmaß eingeführt:

$$LC_{mean}(n) = \frac{1}{N_F} \sum_{j=n-\frac{N_F}{2}}^{n+\frac{N_F}{2}-1} \Delta S_{mean}(j) \qquad (3.28)$$

wobei sich $\Delta S_{mean}(j)$ aus dem Score-Mittelwert aller Konkurrenzmodelle berechnet:

$$\Delta S_{mean}(j) = S_{\hat{m},\hat{s}}(\boldsymbol{x}_j) - \frac{1}{N_S - N_S^{\hat{m}}} \sum_{m\neq\hat{m},s} S_{m,s}(\boldsymbol{x}_j) \qquad (3.29)$$

Analog zur Berechnung des lokalen Scores werden die beiden Maße durch Filterung mit einem Running-Average Fenster der Breite N_F gebildet.

3.4.3 Untersuchungen zu den lokalen Maßen LSR und LAS

Der Frage, der schwerpunktmäßig in diesem Abschnitt nachgegangen werden soll, ist, wie die Abhängigkeit der Modellierungsgenauigkeit von der lokalen Sprechgeschwindigkeit LSR beschaffen ist. Neben Martinez [Mar98] und Morgan [Mor97] konnte insbesondere Pfau [Pfa98a, Pfa99, Pfa00b] zeigen, dass die Worterkennungsrate maßgeblich durch die Sprechgeschwindigkeit beeinflusst wird. Pfau äußerte dabei in [Pfa00a] sowie [Pfa00b] die Vermutung, dass ein Grund hierfür in mangelnder Modellierung der Merkmale bzw. Merkmalsvektoren zu suchen ist. Laut Pfau scheint die Kontextreichweite bei der Delta(-Delta) Berechnung zu einer komplexen, schwer zu modellierenden Vielfalt der Mustervektoren zu führen. Er nimmt hierbei an, dass bei schneller Sprache weitreichendere und damit andere Lautkontextbereiche durch das Deltaraster erfasst werden, als dies bei langsamer Sprache der Fall ist. Er hat versucht, diesen angenommenen Effekt durch eine 'Sprechratennormalisierung' zu kompensieren.

An dieser Stelle steht die Untersuchung der quantitativen Abhängigkeit der Qualität der akustischen Modellierung im Vordergrund. Die Qualität der Modellierung kann in erster Linie am akustischen Score selbst abgeschätzt werden. Idealerweise sollte dieser unabhängig von der Lautdauer gleich gute Ergebnisse liefern. Der Scoreverlauf der Beispieläußerung in Abb. 3.10 zeigt jedoch ein anderes Ergebnis. Im mittleren Graph ist für jeden Frame des Beispielturns

'g091a000' der Score des, anhand der Viterbi-Segmentierung, zugeordneten Modellzustands aufgezeichnet. Auffallend ist einerseits die starke Streuung der ermittelten logarithmierten Wahrscheinlichkeiten, sowie andererseits insbesondere das 'lokale' Verhalten (LAS). Für die Bestimmung des LAS wird die Fensterbreite im folgenden ebenfalls zu $N_F = 100$ Frames angenommen. In Abb. 3.10 ist der Verlauf des LAS (unterer Graph) für den Beispielturn aufgezeichnet. Zur Erhöhung der Anschaulichkeit ist - für die graphische Darstellung - das Vorzeichen von Score und LAS negiert.

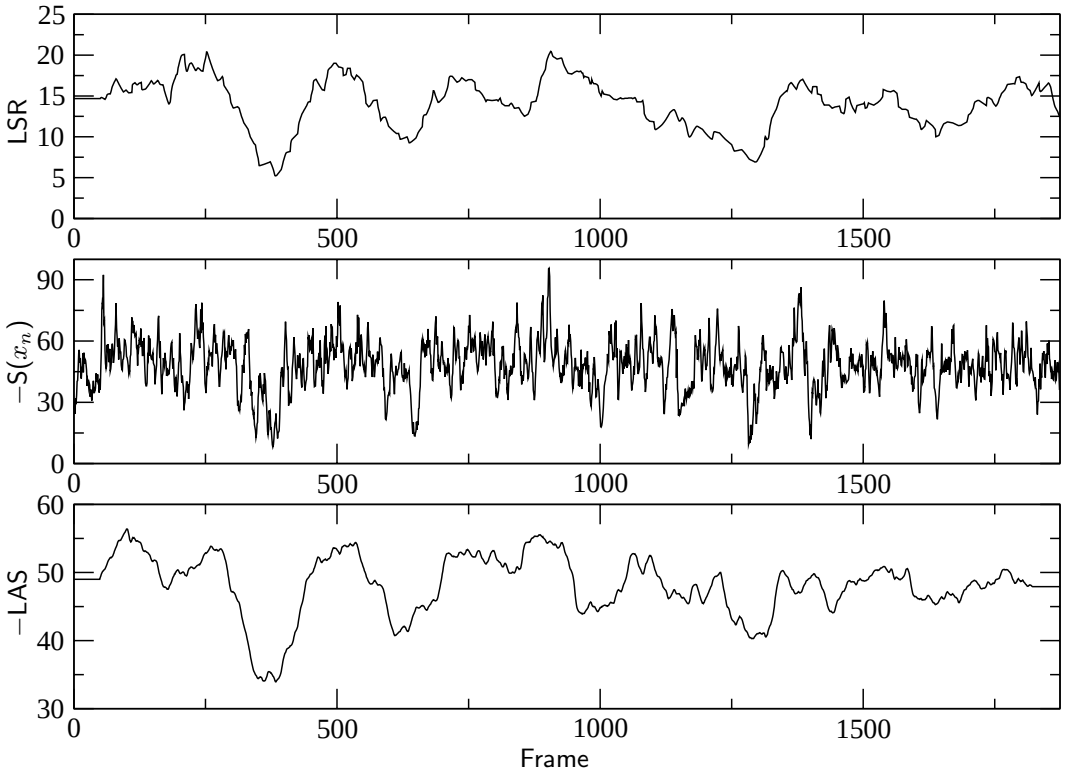

Abb. 3.10: Verlauf von LSR (oben), Score und LAS (unten) des Beispielturns.

Zum Vergleich ist in Abb. 3.10 noch der Verlauf der LSR (oben) angegeben. Deutlich zu Erkennen sind der starke Anstieg des Scores im Bereich langsamer Sprache, sowie ein entsprechender Abfall im Bereich schneller Sprache. Der Korrelationskoeffizient zwischen LSR und LAS für die angegebene Äußerung ergibt sich zu $\rho(LSR, LAS) = \rho_{las} \approx -0.52$. Die Abhängigkeit der Modellierungsgenauigkeit von der Sprechgeschwindigkeit kann also als ziemlich ausgeprägt beschrieben werden. Eine Auswertung des Korrelationskoeffizienten ρ_{las} von ca. 1800 Einzelturns (Verbmobil CD1) führt zu dem in Abb. 3.11 dargestellten Streuplot. Für die rein graphische Darstellung wurde die Zahl der Messpunkte im Verhältnis 20:1 reduziert ('downsampling'). Die Auswertung wurde bewusst auf Daten durchgeführt, die mit zum Training der akustischen HMM-Modelle eingesetzt wurden. Dies erlaubt die Analyse, inwieweit Sprechgeschwindigkeitseffekte bereits in die Modellierung eingehen.

Zur Scoreberechnung wurden Monophonmodelle mit insgesamt ca. 8500 Basisfunktionen

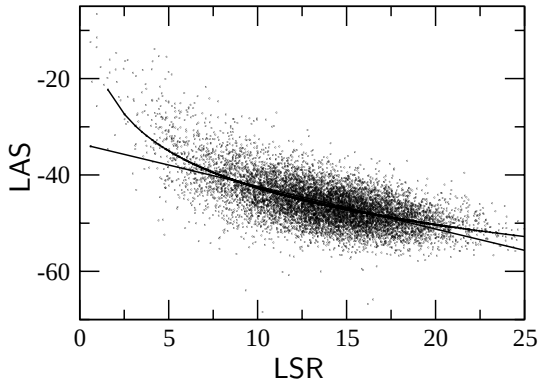

Abb. 3.11: Streuplot der LAS-Werte von CD1 über den zugehörigen LSR-Werten. Die Korrelation beträgt $\rho_{las} = -0.64$

eingesetzt (s. Tab. 5.1). Pausensegmente und andere nichtsprachliche Abschnitte wurden vor der Analyse entfernt. Das Histogramm zeigt, dass LSR und LAS stark negativ korreliert sind. Ein Ansteigen der Sprechgeschwindigkeit führt zu einem Absinken des Scores. Der Korrelationskoeffizient der Punktverteilung in Abb. 3.11 ergibt sich zu $\rho_{las} = -0.64$. Der Streuplot zeigt jedoch auch, dass die Annahme einer linearen Abhängigkeit, speziell im Bereich langsamer Sprechgeschwindigkeit, nur bedingt zutrifft. Zum Vergleich ist eine logarithmische Regressionskurve eingezeichnet.

3.4.3.1 Abhängigkeit der Korrelation vom verwendeten Merkmalsvektor

Die Modelle der vorhergehenden Auswertungen bauen auf den MFCC42-Merkmalsvektoren auf, die neben den statischen Komponenten (hier: 12 MFCC + Energie + Nulldurchgangsrate) auch Geschwindigkeits- und Beschleunigungskomponenten (1. und 2. Ableitung, Δ bzw. $\Delta\Delta$) enthalten. Um zu untersuchen, durch welche Komponenten die Scoreabhängigkeit primär verursacht wird, wurden obige Versuche mit verschiedenen weiteren Modellen wiederholt.

- nur statisch: 12 MFCC + Energie
- statisch+Δ
- statisch+Δ+$\Delta\Delta$

Um einen Vergleich der Auswirkungen auf die Modellierung zu ermöglichen, wurden für die verschiedenen Merkmalsvektorzusammensetzungen eigene Modelle trainiert. Für diese wurde ihre Performanz anhand des erzielten Scores auf dem Trainingskorpus ausgewertet. Zur Auswertung wurde für jede der Äußerungen der satzweise Korrelationskoeffizient zwischen LSR und LAS berechnet. Abb. 3.12 zeigt eine Übersicht über die sich ergebenden Verteilungen der Korrelationskoeffizienten.

Die Histogramme machen deutlich, dass der Einfluss der Sprechgeschwindigkeit mit dem im Merkmalsvektor erfassten Zeitbereich zunimmt. Aufgrund der stark einseitigen Schieflage ('skew') der Verteilungen ist die Annahme einer Normalverteilung zur Beschreibung

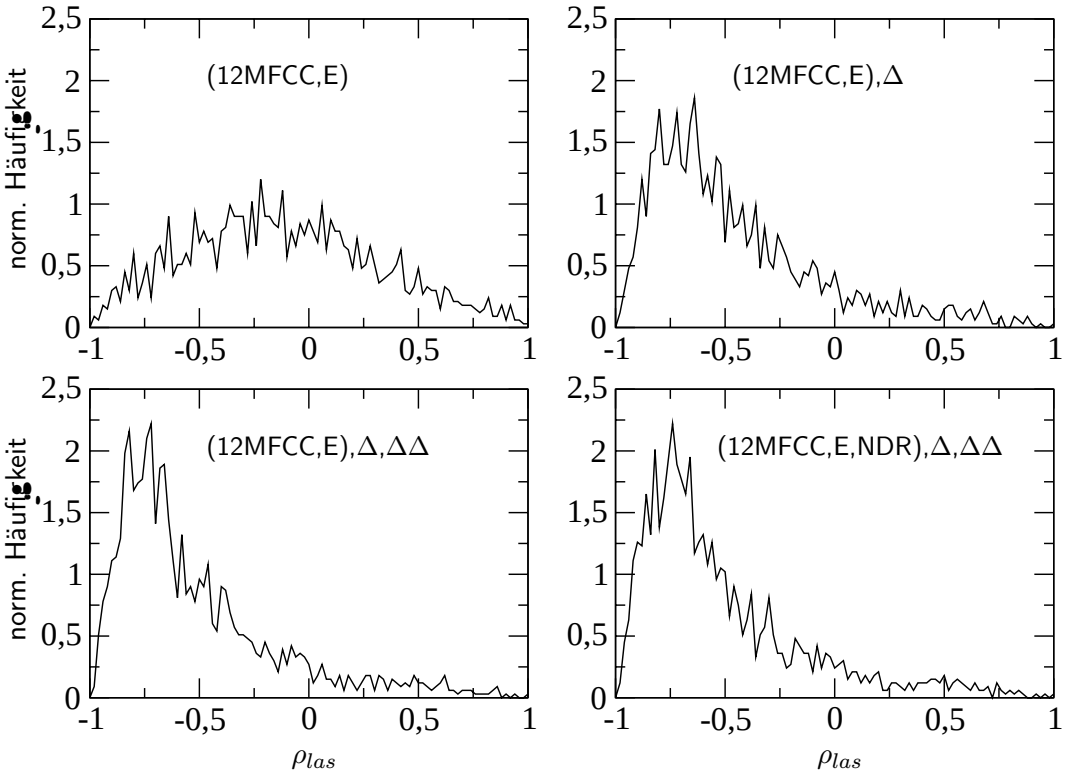

Abb. 3.12: Verteilung des satzweisen Korrelationskoeffizienten ρ_{las} bei unterschiedlichem Aufbau des Merkmalsvektors.

ungenügend geeignet. Eine bessere Aussage liefert der Median-Korrelationskoeffizient. Er beträgt $\rho_{las}^{med} \approx -0.16$ bei den rein statischen Merkmalen. Bei Hinzunahme der 1.Ableitung (Δ) steigt er betragsmäßig bereits auf $\rho_{las}^{med} \approx -0.57$. Wird darüber hinaus noch die 2. Ableitung ($\Delta\Delta$) eingesetzt, so nimmt er nochmals auf $\rho_{las}^{med} \approx -0.65$ zu. Den Einfluss der Sprechgeschwindigkeit ist zwar bei den rein statischen Merkmalen erwartungsgemäß am geringsten, jedoch ist auch hier bereits eine Einwirkung messbar.

Eine Erklärung für das Absinken der Likelihood bei schnellerer Sprechgeschwindigkeit kann in der steigenden Kontextabhängigkeit der Merkmalsvektoren gesehen werden. Bei langsamer Sprechgeschwindigkeit, d.h. bei langen Phonemdauern, liegt die Mehrzahl der Mustervektoren eines Lauts im Innern des Lauts und nicht an dessen Rändern. Als 'Rand' soll hier der Bereich verstanden werden, in dem Nachbarlaute durch die Delta-Berechnung angeschnitten werden. Der eher stationäre Kernbereich weist jedoch vergleichsweise wenig Änderung auf - aufgrund der Mehrzahl an ähnlichen Vektoren, und dementsprechend höheren Wahrscheinlichkeits"masse", wird die Likelihood für diese Vektoren entsprechend groß. An den Rändern jedoch werden die Merkmalsvektoren durch die benachbarten Laute beeinflusst, was sich maßgeblich in der Delta-Berechnung niederschlägt. Der bei der Delta-Berechnung erfasste Kontextbereich beträgt, je nach Aufbau der Vorverarbeitung, bis zu 9 Frames (s. Gl. 1.12). Bei sehr schneller Sprache sinkt die mittlere Lautdauer auf sehr wenige Spektren, was dazu führen kann, dass bei der Delta-Berechnung Spektren mehrerer benachbarter Phoneme ein-

bezogen werden. Da als Kontext im Prinzip nahezu beliebige Lautkombinationen auftreten können, steigt die Variation in den Merkmalsvektoren stark an. Bei der Modellierung mit einer konstanten Anzahl von Verteilungen sinkt damit die Likelihood eines einzelnen Vektors erheblich - verglichen mit der Modellierung der Vektoren aus dem Kernbereich eines Phonems.

Die Experimente dieses Abschnitts mit Merkmalen ohne Ableitungskoeffizienten darauf hin, dass - wenn auch vergleichsweise schwach - bereits die rein statischen Merkmale durch die Sprechgeschwindigkeit beeinflusst werden. Analog zur Argumentation von Pfau [Pfa00b], könnte hier vorgebracht werden, dass eine konstante Fensterbreite, wie sie bei der Frameweisen FFT zur Fensterung des Zeitsignals eingesetzt wird, nicht geeignet ist, um die sprechgeschwindigkeitsbedingten Veränderungen genügend zu erfassen. Seiner Ansicht nach ist die 'statische' Frameraster bei Delta-Berechnung dafür verantwortlich, dass in Abhängigkeit von der Sprechgeschwindigkeit unterschiedliche Kontextbereiche einbezogen werden. Sein Vorschlag zur Kompensation war eine 'Sprechratennormalisierung'. Kern dieses Ansatzes ist eine dynamische Neuausrichtung des Framerasters durch Interpolation. Fehlende Spektren (rein statische Komponenten) werden durch Interpolation hinzugefügt, wobei auch die vorhandenen Spektren einbezogen und verändert werden. Längere Laute werden entsprechend verkürzt. Ähnliche Vorgehensweisen finden sich in [Tsu00] und [Ned01].

Den Ansätzen ist das Konzept einer *Normalisierung* gemein, d.h. längenmäßig abweichende Laute werden auf eine mittlere Normdauer bzw. Normsprechgeschwindigkeit korrigiert. Die Analyse der Scoreabhängigkeit wirft jedoch die Frage auf, inwieweit eine *Normalisierung auf mittlere Sprechgeschwindigkeit* überhaupt sinnvoll ist. Dies betrifft insbesondere die Normalisierung langsamer Abschnitte. Der Streuplot in Abb. 3.11 zeigt steigende Scores bei fallender Sprechgeschwindigkeit, entsprechend für langsame Sprache die höchste Übereinstimmung der Mustervektoren mit den zugehörigen stochastischen Modellen. Diese Aussage wird im folgenden anhand der Konfidenzmaße aus Abschnitt 3.4.3.3 nochmals bestätigt. Die Normalisierung führt zu einer Verkürzung der langsamen Abschnitte und dementsprechend zu einem Verwerfen aussagekräftiger Information. Sie kann vor diesem Hintergrund daher nur als bedingt sinnvoll betrachtet werden.

Dem widersprechen allerdings in gewissem Umfang die Ergebnisse von Nedel und Stern [Ned01]. Die Autoren untersuchten die Dauernormalisierung für verschiedene Zielphonemlängen im Bereich von 6..15 Frames. Es zeigte sich im gesamten Bereich eine Verbesserung gegenüber der Basisfehlerrate, wobei das Optimum bei etwa 8..10 Frames normalisierter Länge erzielt wurde. Relativierend müssen zwei Punkte angeführt werden. Die Autoren bewerten die Performanz der Normalisierung anhand der WER, machen jedoch keine Angaben darüber, inwieweit weitere Parameter - hier insbesondere das LM-Gewicht (vgl. Abschnitt 4.2.1) - mit optimiert wurden. Darüber hinaus handelt es sich bei dem vorgestellten Verfahren um einen Ansatz zur *Phonemdauer*-Normalisierung. Da *alle* Phoneme auf die *gleiche* gemeinsame Länge normalisiert werden, sind hierdurch die verschiedenen Phoneme zwangsläufig unterschiedlich stark betroffen. Gerade Plosivlaute, die ja im Mittel kürzer als beispielsweise Vokale sind, werden daher bei einer langen Zieldauer stark verzerrt.

Anders als bei langsamer Sprache sollte eine Normalisierung, d.h. Streckung, bei schneller Sprache jedoch gute Erfolge zeigen. Wie die Histogramme in Abb. 3.12 zeigen, wird die Sprechgeschwindigkeitsabhängigkeit erst voll durch die Einführung und Verwendung der Ableitungskoeffizienten wirksam. Durch sie wird die Variabilität der Merkmalsvektoren in den zeitlichen Phonemgrenzbereichen stark erhöht. Durch eine Streckung sollte die Variabilität reduziert werden können.

Die Betrachtung der Scoreabhängigkeiten wurde mit kontextunabhängigen Monophonmodellen durchgeführt. Interessant ist in diesem Zusammenhang das Verhalten von kontextabhängigen Einheiten. Untersucht wurde dies anhand von Triphonmodellen, die vom rechts- bzw. linksseitigen Nachbarphonem abhängen.

3.4.3.2 Abhängigkeit der Korrelation bei kontextabhängiger Modellierung

Die für die Untersuchung verwendeten Triphonmodelle wurden mittels eines Entscheidungsbaumverfahrens [Bah91]) (s. Kap. 4) erzeugt. Die Zahl der Zustände nach State-Tying beträgt ca. 2500, die Zahl der eingesetzten Normalverteilungen umfasst ca. 35000 (s. Tab. 5.1). Bezüglich der Erkennungsrate sind diese Modelle leistungsfähiger als vergleichbare Monophonmodelle. Kontextabhängige Modelle sind darauf ausgelegt, die vom phonetischen Kontext abhängigen Variationen, die sich in veränderten Spektren niederschlagen, individuell und damit exakter zu modellieren. So zeigen beispielsweise, die in dieser Arbeit verwendeten Basistriphonmodelle (s. Tab. 5.1), auf dem Eval96-Testset eine ca. 5% absolut geringere Wortfehlerrate. Ein leicht differenziertes Bild ergibt sich jedoch bei Betrachtung der Sprechgeschwindigkeitsabhängigkeit der Modellierungsgenauigkeit. Abb. 3.13 zeigt die Verteilung der Korrelationskoeffizienten zwischen lokaler Sprechgeschwindigkeit und lokalem Score.

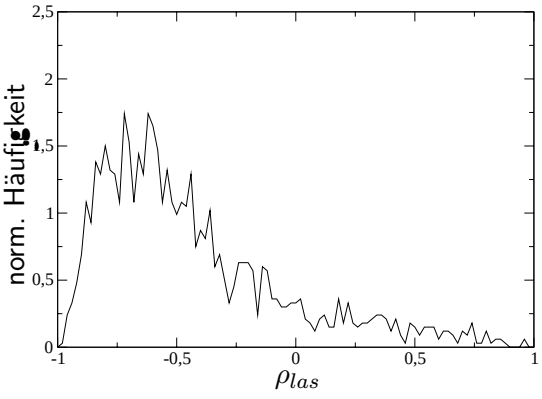

Abb. 3.13: Histogramm über die ermittelten Korrelationskoeffizienten ρ_{las} bei Verwendung von Triphonmodellen. Der Median-Korrelationskoeffizient beträgt $\rho_{las}^{med} = -0.57$.

Erstaunlicherweise ergibt sich auch bei Triphonmodellen eine relativ hohe Abhängigkeit des LAS von der lokalen Sprechgeschwindigkeit. Dies zeigt, dass die kombinatorische Vielfalt, die sich bei schneller Sprache bedingt durch das Frameraster ergibt, durch Triphonmodelle ebenfalls nur bedingt abgedeckt werden kann.

3.4.3.3 LSR und Konfidenzsicherheit

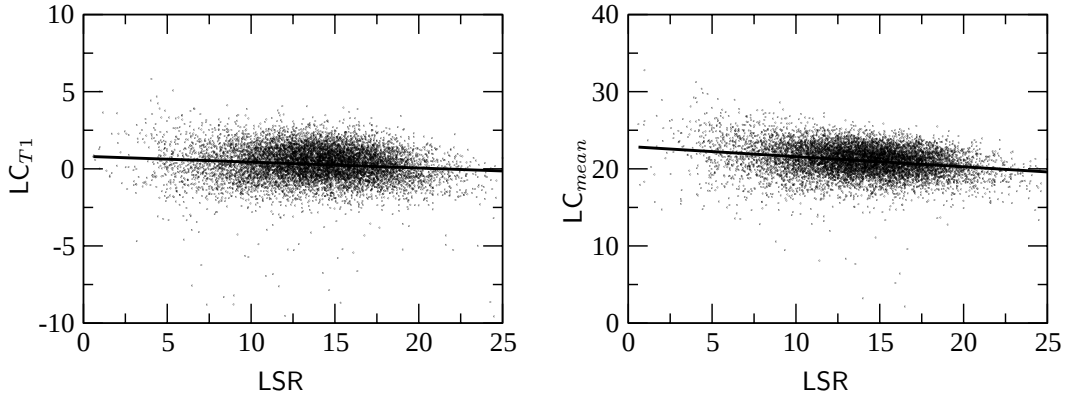

Abb. 3.14: LC_{T1} und LC_{mean} Verteilung der Trainingsdaten von CD1. $\rho_{LC_{T1}} = -0.11$ und $\rho_{LC_{mean}} = -0.25$.

Abb. 3.14 zeigt die Verteilung zwischen LC_{T1} und LC_{mean} und LSR. Die Verteilung von LC_{T1} (links) weist eine relative Unabhängigkeit von der Sprechgeschwindigkeit auf. Der Korrelationskoeffizient beträgt ca. -0.11. Die Modelle von konkurrierenden, stark ähnlichen Lauteinheiten scheinen also der gleichen Verschlechterung unterworfen zu sein. Etwas anders sieht das Verhalten von LC_{mean} aus. LC_{mean} beschreibt den Abstand des korrekten zum mittleren Score der Konkurrenzmodelle. Mit einem Korrelationskoeffizienten $\rho_{LC_{mean}} = -0.25$ ist die Abhängigkeit zwar geringer als bei LAS - aber die negative lineare Abhängigkeit zeigt dennoch, dass bei steigender Sprechgeschwindigkeit die Diskriminanz zwischen korrekten und konkurrierenden Modellen sinkt. Dies geschieht bereits bei der stochastischen Modellierung *der Trainingsdaten.*

3.4.3.4 Prädiktion der Sprechgeschwindigkeit

Die im vorangegangenen Abschnitt beschriebenen Untersuchungen bezogen sich auf die Auswirkungen der Modellierung, wie sie bereits im bzw. auf dem Trainingsmaterial beobachtet werden können. Es wurde festgestellt, dass für die Trainingsdaten eine starke Korrelation zwischen lokaler Sprechgeschwindigkeit und mittlerem, laufendem Score zu erkennen ist. Im weiteren Verlauf wurde untersucht, inwieweit sich diese Information dazu verwenden lässt, bei unbekannten Sprachdaten von der aktuellen Scoreberechnung auf die Sprechgeschwindigkeit zu schließen bzw. diese sogar zu prädizieren.

Der globale Zusammenhang zwischen LSR und LAS auf den Trainingsdaten kann Abb. 3.11 entnommen werden. Die aus dieser Verteilung gewonnene Regressionsgerade wurde in Abb. 3.15 auf die Testdatenverteilung übertragen. Wird aufgrund dieser Punktverteilung eine Regressionsgerade ermittelt, so ergibt sich bei einer linearen Regression:

$$LAS = -33.509 - 0.88507 LSR \tag{3.30}$$

Wird die logarithmische Regression zugrundegelegt, so folgt hieraus:

$$LAS = -17.215 - 11.047ln(LSR) \tag{3.31}$$

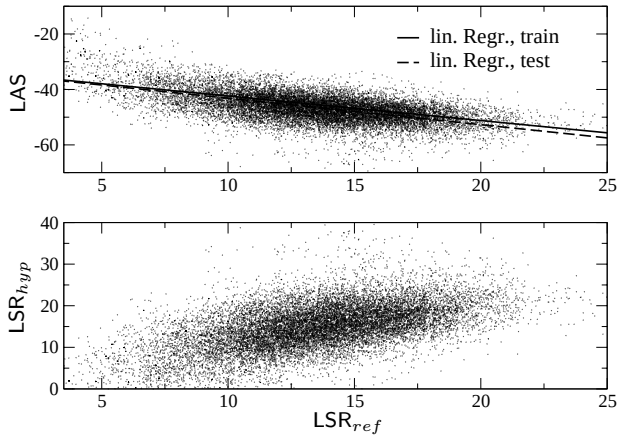

Abb. 3.15: Streuplot zwischen LAS und LSR auf den Eval96-Testdaten (oberes Bild). Die untere Abbildung zeigt die Verteilung der hypothetisierten LSR versus die Referenz-LSR. Die Hypothese ergibt sich durch die Umkehrung der Regression.

Aus Gründen der Darstellbarkeit, wurde in obiger Abbildung das Datenaufkommen durch Downsampling auf 10% des ursprünglichen reduziert. Bei Ermittlung des globalen Korrelationskoeffizienten aufgrund dieser 'globalen' Streuung ergibt sich ein Wert $\rho_{las} \approx 0.60$. Wie diese Regressionen zur Extrapolation der Sprechgeschwindigkeit eingesetzt werden kann, wurde anhand des Eval96 Testsets, das keine Sprecher der Trainingsdatenbank enthält, untersucht. Der untere Graph in Abb. 3.15 zeigt einen Streuplot zwischen der anhand der linearen Regression geschätzten Sprechgeschwindigkeit auf dem Evaluierungsmaterial und der Referenz-LSR. Die Referenz-LSR wurde hier mittels einer Forced-Viterbi Segmentierung anhand der Referenztranskription ermittelt. Der Streuplot macht deutlich, dass die direkte Übertragung der Regression zu einer überhöhten Schätzung führt. Letzteres ist mit der Diskrepanz zwischen Trainings- und Testmaterial zu erklären, die modellierungsbedingt zu schlechteren Scores führt. Diese Abweichung lässt sich jedoch durch eine Mittelwertkorrektur reduzieren: Aus einem Langzeitmittelwert des Evaluierungsscores kann die Abweichung zum Langzeitmittelwert der Trainingsdaten ermittelt werden. Die Regressionsgerade kann dann um diese Abweichung korrigiert werden. Die Korrelation zwischen Prädiktion und Referenz beträgt etwa 0.60 und hat damit annähernd die Ausprägung wie auf dem Trainingsmaterial.

3.4.3.5 Untersuchungen zur Sprechgeschwindigkeitskompensation mittels klassenspezifischer Modelle

Der Einsatz klassenbasierter Sprechgeschwindigkeitsmodelle ist in der Literatur von einigen Autoren aufgegriffen worden: von den meisten wird vorgeschlagen, das Trainingsmaterial in drei oder mehr Klassen, z.B. langsam, mittel und schnell, einzuteilen und für jede der Klassen individuelle HMM-Modelle zu trainieren bzw. abzuleiten. Von den Autoren wird diese Einteilung meist auf Satzebene vorgenommen. Zu unterscheiden ist hier zwischen Trainings-

und Erkennungs(=Auswahl)phase. Für die Auswahl der Modelle während der Erkennungsphase wurden zwar schon Einteilungen auf Sub-Satzebene vorgeschlagen - so versuchte Pfau die Klasseneinteilung individuell für Spurts. Diese Feineinteilung bleibt jedoch auf die Erkennung beschränkt.

Die Untersuchungen zur lokalen Sprechgeschwindigkeit haben bereits aufgezeigt, dass die Sprechgeschwindigkeit während einer Äußerung starken Schwankungen unterworfen sind - insbesondere bei spontansprachlichen Sprachkorpora. Um die Kapazität klassenbasierter Modelle betreffend ihrer Adaptionsleistung zu bewerten, können LSR und LAS herangezogen werden. Beispielhaft sei dies an MAP-trainierten Klassenmodellen dargestellt. Die Klassen- bzw. Dateneinteilung in die 3 Geschwindigkeitsgruppen wurde anhand eines satzweisen, sowie eines lokalen (LSR) Sprechgeschwindigkeitsmaßes vorgenommen, wobei die Klassengrenzen in beiden Fällen zu $\mu_{ROS_{Ph}} \pm \sigma_{ROS_{Ph}}$ festgelegt wurden. Die jeweiligen Klassenmodelle wurden mittels MAP-Nachschätzung aus einem generischen Basismodell abgeleitet. Die Scoreberechnung mit jedem der zwei 3er Sets bildet die Basis für die Bestimmung des LAS-Verlaufs. Analog zu Abb. 3.11 kann anhand der ermittelten LSR-zu-LAS Verteilungen für jede Klasse eine Regressionsgerade bestimmt werden. Die drei Regressionsgeraden sind in Abb. 3.16 wiedergegeben.

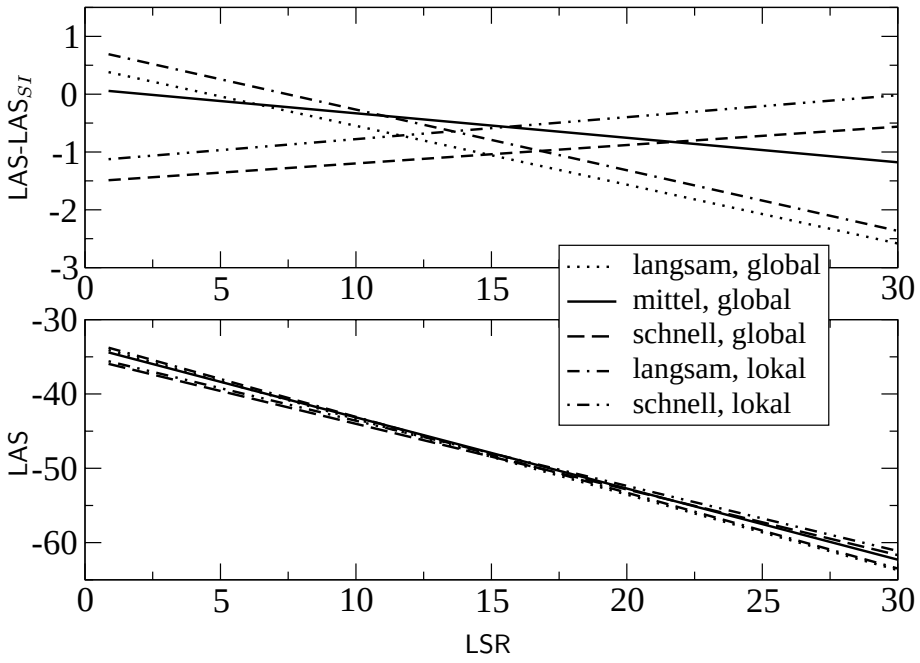

Abb. 3.16: Regressionsgeraden für langsame, mittlere und schnelle Modelle basierend auf globaler und lokaler Sprechgeschwindigkeitskategorisierung.

Die Kurven zeigen zwei prinzipielle Ergebnisse. Einerseits hat das Training sprechgeschwindigkeitsspezifischer Modelle - auch bei globaler Einteilung - tatsächlich den gewünschten Effekt. So weisen die für mittlere Sprechrate ausgelegten Modelle auch in diesem Bereich die beste Performanz auf. Für schnelle bzw. langsame Sprache andererseits dominieren - wie

beabsichtigt - die schnellen, respektive langsamen Modelle. Offensichtlich jedoch fallen diese in den jeweiligen Bereichen erzielten Verbesserungen, im Vergleich zum sprechgeschwindigkeitsbedingten Abfall, sehr gering aus.

3.5 Phonemdauer und Score

In den bisherigen Untersuchungen konnte gezeigt werden, dass der durchschnittliche Score eng mit der jeweiligen aktuellen Sprechgeschwindigkeit verknüpft ist. Die Sprechrate berechnet sich im Prinzip jedoch aus dem Kehrwert der Phonemdauer(n) (s. Gl. 3.1 bzw. 3.2). Im folgenden soll die Abhängigkeit zwischen Phonemdauer und aktuellem Score näher untersucht und diskutiert werden. In der folgenden Abbildung 3.17 ist für die Phonemdauern von 1 bis 50 Frames der jeweilige mittlere Score sowie die zugehörige Standardabweichung dargestellt. Im linken Diagramm wurde die MFCC42-Vorverarbeitung zugrundegelegt, wohingegen im rechten Diagramm die aus ausschließlich statischen Merkmalen bestehende MFCC12-Vorverarbeitung eingesetzt wurde.

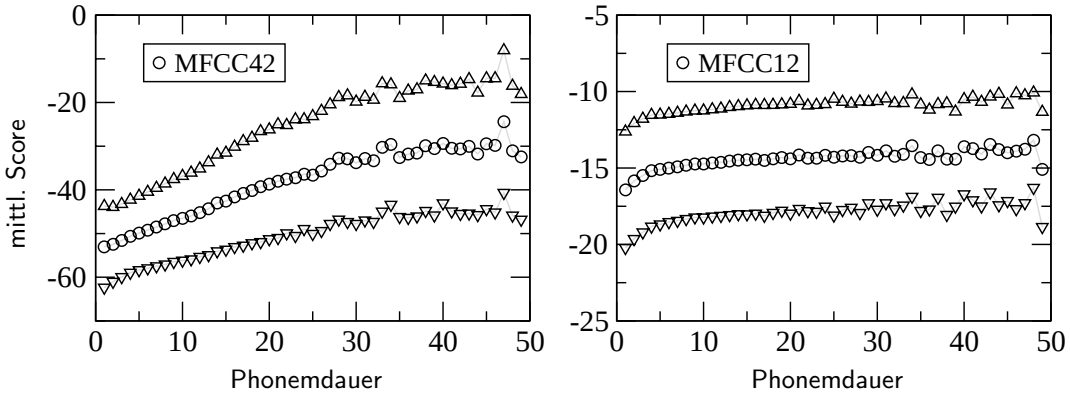

Abb. 3.17: Abhängigkeit zwischen Phonemdauer T_{Ph} und mittlerem Score $\bar{S}_{T_{Ph}}$ bei MFCC42-Vorverarbeitung (links) und MFCC12-Vorverarbeitung (rechts). Je Dauer sind Mittelwert $\mu_{T_{Ph}}$ ($\circ$) und $\mu_{T_{Ph}} \pm \sigma_{T_{Ph}}$ (∇,$\triangle$)dargestellt. Im Bereich 1..30 Frames beträgt die Korrelation zwischen Phonemdauer und mittlerem Score je Dauer 0.99.

Im linken Diagramm zeigt sich für den Bereich 1..30 Frames eine hohe Korrelation zwischen mittlerem Score und Phonemdauer. Der Korrelationskoeffizient erreicht einen Wert von $\rho_{\bar{S}}^{T_{Ph}} = 0.99$. Die logarithmierte Modellwahrscheinlichkeit steigt also im Mittel nahezu direkt proportional mit der Länge der Phoneme. Bei rein statischen Merkmalen (rechter Graph) ist eine stärkere Abhängigkeit zwischen Dauer und mittlerem Score nur bei einer Phonemlänge von weniger als 5 Frames bemerkbar, darüber scheinen beide Merkmale nahezu unkorreliert zu sein.

Mit zunehmender Länge steigt der Anteil an Frames, der im Kernbereich eines Phonems liegt. Je kürzer ein Phonem ist, desto größer ist der Anteil solcher Frames, die durch benachbarte Phoneme beeinflusst werden. Bei langen Phonemen, d.h. mehr als 9 Frames (bzgl. der implementierten Vorverarbeitung inkl. Delta-Berechnung), erfasst das gesamte Delta-Raster

nur Frames des *eigenen* Lauts. Die Varianz ist dementsprechend sehr niedrig. Bei den Frames, die direkt an an einer Phonemgrenze liegen, greift das Deltaraster direkt in das benachbarte Phonem. Ist dieses nur ein oder zwei Frames breit, so erfasst das Deltaraster sogar mehr als den unmittelbaren Nachbarlaut. Für den Fall, dass das betrachtete, zentrale Phonem selbst nur ein oder zwei Frames breit ist, betrifft dies sowohl den rechten als auch den linken Kontext. Mit zunehmender Länge der Phoneme wird der mittlere Score unabhängiger von der phonetischen Nachbarschaft. Dies äußert sich in der Unabhängigkeit des mittleren Scores von der Phonemlänge, wie sie oberhalb von 30 Frames beobachtbar ist.

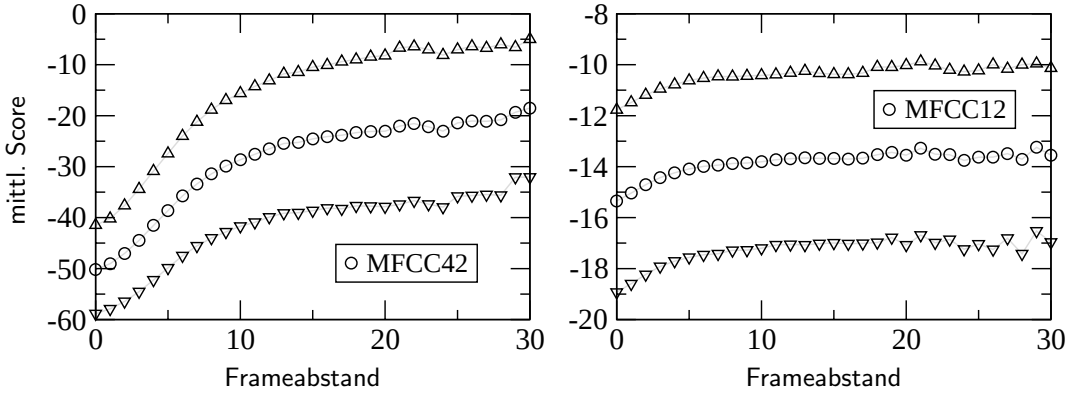

Abb. 3.18: Abhängigkeit zwischen Frameabstand zu Phonemgrenze $d_b^r(n)$ und zugehörigem mittlerem Score $\bar{S}_{\Delta T}$: starker Anstieg des Scores bis zu ca. 10 Frames, darüber annähernd konstanter Verlauf (rechts: MFCC42, links: MFCC12).

Eine Bestätigung dieser Beobachtung liefern die Diagramme (rechts: MFCC42, links: MFCC12) in Abbildung 3.18. Sie geben den mittleren Score $\bar{S}_{\Delta T}$ je Abstand $d_b^r(n)$ eines Frames n zur nächstgelegenen Segmentierungsmarke wieder. Für jeden Frame einer Äußerung r wurde hierzu der Abstand individuell bestimmt:

$$d_b^r(n) = \min_{i=1\ldots N_B^r} |n - b_i^r| \tag{3.32}$$

Für jeden möglichen Abstandswert j wird der mittlere Score berechnet:

$$\bar{S}_j = \frac{\sum_{r=1}^{R}\sum_{n=1}^{T_r} S_{m,s}(n)\delta_b^r(n)}{\sum_{r=1}^{R}\sum_{n=1}^{T_r}\delta_b^r(n)} \tag{3.33}$$

$$\delta_b^r(n) = \begin{cases} 1 & \text{wenn } d_b^r(n) = j \\ 0 & \text{sonst} \end{cases}$$

Pausensegmente und andere nichtsprachliche Bereiche wurden vorher eliminiert. Zusätzlich wurde noch für jeden Abstandswert die jeweilige Scorevarianz berechnet. In Abbildung 3.18 zeigt die mittlere Punktereihe die Scoremittelwerte $\bar{S}_j$. Die obere und untere Punktereihe geben die zugehörige Abweichung $\bar{S}_j \pm \sigma_j$ wieder. Der Mittelwertverlauf weist einen deutlichen Anstieg bis zu einem Bereich von etwa 10 Frames auf, danach flacht der Anstieg sehr stark ab.

Normiert man den mittleren Score $\bar{S}_j$ auf die Dimension N_D des eingesetzten Merkmalsvektors, so ergeben sich die Verläufe in Abb. 3.19. Die Normierung kann unter der Annahme, dass die Emissionswahrscheinlichkeit primär durch jeweils eine Gaussverteilung eines Zustands bestimmt ist, gerechtfertigt werden [Kan00]. Bei Gaussverteilungen mit diagonalen Kovarianzen kann die Berechnung der Likelihood, aufgrund der Unabhängigkeit der Einzeldimensionen, auch komponentenweise erfolgen [Boc01].

$$\begin{aligned}\log \mathcal{N}(\boldsymbol{x}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) &= \log \frac{1}{\sqrt{(2\pi)^{N_D}|\boldsymbol{\Sigma}|}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})} \\ &= \log \prod_{i=1}^{N_D} \frac{1}{\sqrt{(2\pi)\Sigma_{ii}}} e^{-\frac{1}{2}(x_i-\mu_i)^2/\Sigma_{ii}} = \sum_{i=1}^{N_D} \log \mathcal{N}(x_i, \mu_i, \Sigma_{ii}) \qquad (3.34)\end{aligned}$$

Auffallend bei den Scoreverläufen in Abb. 3.19 ist, dass, trotz des starken Abfalls bei kurzen Phonemdauern, die Vorverarbeitung mit $\Delta\Delta$-Koeffizienten im ganzen Bereich einen höheren mittleren Score je Dimension aufweist.

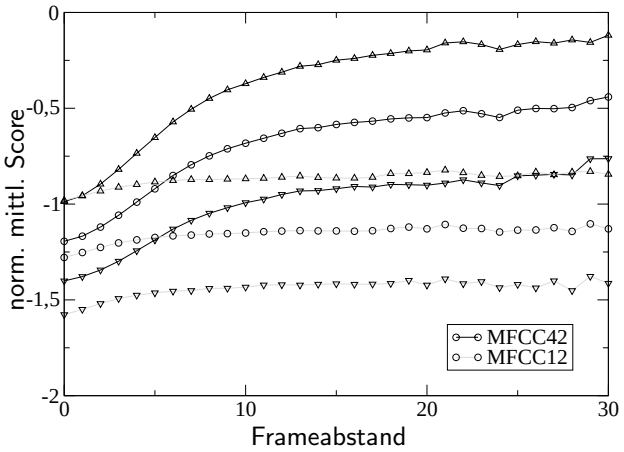

Abb. 3.19: Abhängigkeit zwischen Frameabstand zu Phonemgrenze $d_b^r(n)$ und zugehörigem normierten mittleren Score $\frac{1}{N_D}\bar{S}_{\Delta T}$. Vorverarbeitung mit DeltaDelta-Merkmalen liefert auch bei sehr kurzen Phonemdauern noch höhere normierte Scores.

Über die Gleichungen 3.1 bzw. 3.2 lässt sich von der mittleren Phonemdauer auf die Sprechgeschwindigkeit schließen. Die Diagramme 3.17-3.18 bestätigen, dass durch die Einführung der Delta-Koeffizienten eine deutliche Abhängigkeit der akustischen Modellierung von der vorliegenden Sprechgeschwindigkeit induziert wird. Mit fallender Sprechgeschwindigkeit, d.h. steigender Phonemdauer nimmt die Abhängigkeit ab. Bei kürzerer Lautdauer ist der Anteil an Mustervektoren (bezogen auf die Gesamtlänge) aus dem transienten Bereich zwischen Phonemen jedoch höher. Eine spezialisierte, kontextabhängige Modellierung der möglichen Musterverläufe dieser Bereiche sollte daher von Vorteil sein. Diesbezügliche Verfahren werden im folgenden Kapitel diskutiert.

Kapitel 4

Zustandsgruppierung mit Entscheidungsbäumen

4.1 Kontextmodellierung: Stand der Technik

4.1.1 Einführung

Als Grundeinheiten für den Erkennungsvorgang werden häufig Phoneme angenommen. Phoneme können jedoch nicht einfach als isolierte Einheiten betrachtet werden, da sie in ihrem akustischen Verlauf durch die vorausgehende und nachfolgende phonetische Nachbarschaft beeinflusst werden. Dieser Effekt wird als Koartikulation bezeichnet. Der Grund hierfür ist im Aufbau und der Funktion des menschlichen Sprachorgans, des Vokaltrakts, zu sehen. Das Sprechen, d.h. die Bewegung des Vokaltrakts ist ein dynamischer und kontinuierlicher Vorgang. Sprachliche "Bausteine" können daher nicht einfach hintereinandergesetzt werden. Dies würde zu Brüchen, Unstetigkeiten in der Bewegung führen. Da der Vokaltrakt im Laufe des Lebens - unbewusst - auf eine stetige und insbesondere ökonomische Bewegung gelernt worden ist, kommt es zu Verschleifungen an den Grenzbereichen von Lauten. Die Vokaltraktbewegung wird sowohl entsprechend der vorausgehenden, als auch der nachfolgenden Laute angepasst. Die kontextabhängige Variation der Vokaltraktbewegung äußert sich in spektralen Veränderungen des Sprachsignals und führt somit unmittelbar zu veränderten Mustervektoren. Dies zeigt sich insbesondere bei Merkmalsvektoren aus den transienten Bereichen zwischen Phonemen. Abb. 4.1 verdeutlicht diese Abhängigkeit schematisch für 2-dimensionale Mustervektoren.

Schematisch dargestellt sind Punkte von Musterfolgen. Deren prinzipieller Verlauf wird im rechten Teil der Abbildung abhängig von der nachfolgenden (Laut)Klasse. Skizziert ist hier ein rechtsseitiger Kontext mit Abhängigkeit von einer Nachfolgerklasse $\{B, C\}$. Bei der Sprachproduktion ist dieser weitreichender als nur der direkte Nachbarlaut. Bei einer sogenannten kontextunabhängigen Monophonmodellierung wird genau ein stochastisches Modell (HMM) für den gesamten Laut erzeugt. Die Parameter dieses Modells würden anhand aller Mustervektoren (Bereich A) geschätzt. Bei kontextabhängiger Modellierung wird nach der Reichweite des betrachteten Kontexts unterschieden. Sogenannte Biphon-Modelle betrachten nur den unmittelbaren, wahlweise rechten oder linken Kontextlaut. Bei Triphonmodellen hin-

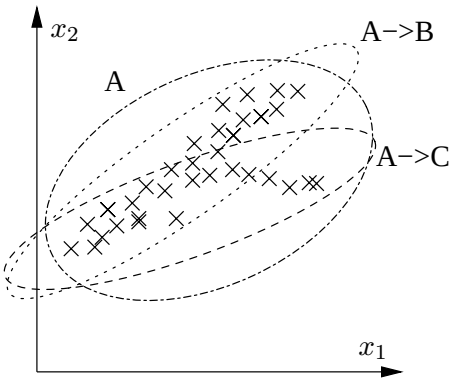

Abb. 4.1: Koartikulation: Verlauf der Musterfolge ist abhängig von den Folgelauten.

gegen werden beide Nachbarlaute berücksichtigt. Weitreichendere Kontextmodelle, wie beispielsweise Quintphone (jeweils zwei Nachbarlaute) werden vorzugsweise bei Systemen zur Nachrichtenverschriftung [Kem00] eingesetzt. Im allgemeinen beschränkt man sich bei der Kontextbetrachtung auf 1-5 Lautkontexte je Seite [Fin97a, Bah91]. Aktuelle Systeme werden i.d.R. mit Triphonmodellen aufgebaut. Im folgenden wird daher die stochastische Modellierung mit Triphoneinheiten näher betrachtet. Zur Beschreibung eines Triphons sei im folgenden die Nomenklatur 'L_X_R' vereinbart. X kennzeichnet hier das zentrale Basisphonem, wohingegen L und R das jeweilige links- bzw. rechtsseitige Kontextphonem charakterisieren.

Die Zahl der zu generierenden Modelle steigt exponentiell mit der Anzahl der berücksichtigten Kontexte. Bei einfachen Triphonen (rechts- und linksseitigem Kontext) gibt es N_{Ph}^3 theoretisch mögliche Kontextkombinationen, wobei N_{Ph} die Zahl der betrachteten Phonemeinheiten angibt. Bei $N_{Ph} = 50$ Phonemen beläuft sich die Zahl der unterschiedlichen Triphone bereits auf 125000. In praktischer Hinsicht reduziert sich die Zahl der zu erfassenden Triphone, da gewisse Lautkombinationen, je nach Sprache, aus phonotaktischen Gründen nicht möglich sind. Jedoch ist es auch bei sehr großen Sprachkorpora äußerst unwahrscheinlich, dass alle Kombinationen auftreten. Darüber hinaus treten viele Kombinationen auch nur in geringem Umfang auf, so dass die Schätzung eines individuellen statistischen Modells nicht robust genug erfolgen kann. So ergibt eine Auswertung des verwendeten Verbmobil-Trainingskorpus (CDs 1-5,7,12) weniger als 10000 vorkommende, unterschiedliche Phonem-Tripel [Fra00].

Aus diesem Grund werden meist nur die häufigen Kombinationen individuell modelliert. Primitive Ansätze treffen hierbei eine Auswahl nur anhand der Auftretensstatistik der Triphone, etwa durch Auswertung der reinen Auftretenshäufigkeit, oder, etwas verfeinert, unter zusätzlicher Berücksichtigung der jeweiligen Triphondauer in Frames. Ein Nachteil dieses Ansatzes ist, dass die Zustände der HMM-Modelle mancher Laute unnötigerweise separat modelliert werden. Zur Erklärung sei hier wieder auf Abb. 4.1 verwiesen. In der linken Hälfte ist der Verlauf der Mustervektoren nahezu unabhängig vom nachfolgenden Kontext. Dieser Bereich könnte vorteilhafter durch eine gemeinsame, parametrische Verteilung beschrieben werden. Eine separate Modellierung - gleichbedeutend mit einer Teilung der Trainingsdaten

- erschwert in diesem Fall nur die robuste Schätzung der HMM-Verteilungsparameter. Die gemeinsame Nutzung von Parametern wird auch als "Tying" bezeichnet. Unter Zustands- oder (engl.) State-Tying wird dementsprechend der Fall verstanden, dass unterschiedliche HMM-Modelle in einem Zustand die Verteilungsparameter gemeinsam verwenden.

Bei der Beschreibung des einfachen, an der Auftretenshäufigkeit ausgerichteten Triphoneinteilung zeigen sich bereits zwei grundlegende Probleme. Für jedes Triphon muss eine Modellrepräsentation gefunden und trainiert werden - d.h. auch für Triphone, die nicht in den Trainingsdaten gesehen wurden. Da in diesem Fall keine Beispieldaten für eine solche Lautkombination vorliegen, muss stattdessen eine Abbildung auf ein Alternativmodell, das als ähnlich oder repräsentativ angesehen werden kann, festgelegt werden. Als zweite Schwierigkeit stellt sich an dieser Stelle das Schätzproblem der Verteilungsparameter. Selbst für die auftretenden Kombinationen liegt häufig nicht genügend Trainingsmaterial vor. Für diese muss ebenfalls eine sinnvolle und robuste Verknüpfung gefunden werden, die ein Parametertraining erlaubt. Zur Lösung dieses Problems wurden primär zwei datengetriebene Ansätze vorgeschlagen:

- Bottom-Up Zustands-Gruppierung
- Entscheidungsbäume (als Spezialfall der Top-Down Zustands-Gruppierung)

Das Ziel ist in beiden Fällen das Zusammenfassen von ähnlichen Zuständen zu Gruppen, die sich durch eine gemeinsame Verteilung repräsentieren lassen. Bottom-Up Clustering kennzeichnet die iterative Zusammenfassung von Einheiten [Mak96, ChC97]. Das agglomerative Zusammenfassen erfolgt vorzugsweise auf Sub-HMM-Ebene, beispielsweise individuell für jeden Zustand der Triphonmodelle.

Im Rahmen dieser Arbeit wurde ein Bottom-Up Clusteralgorithmus zur Integration generalisierter Kontexte realisiert [Fra00]. Da jedoch die Startkondition eines Bottom-Up Verfahrens nur die vorkommenden Triphone berücksichtigen kann, ist ein nicht zu vernachlässigender Nachteil dieser Strategie darin zu sehen, dass nur Zustandsgruppierungen für diejenigen Lautkombinationen gefunden werden können, die in der Trainingsdatenbasis auch wirklich aufgetreten sind. Für die übrigen, ungesehenen Exemplare muss ein Ersatzmodell ("Fall-Back") gefunden werden. Dies ist insbesondere dann von Bedeutung, wenn im Testlexikon Kombinationen auftreten, die im Trainingskorpus nicht angetroffen wurden. Eine Analyse des Verbmobil Eval96 Testsets weist jedoch lediglich 4 Triphone auf [Fra00] auf, die nicht im Trainingsset gesehen wurden. Dies gilt allerdings nur bei Verwendung von Intrawort-Triphonen, bei denen die Kontextinformation nicht über Wortgrenzen hinweg betrachtet wird. Die Wahrscheinlichkeit auf ungesehene Lauttripel zu treffen ist höher, wenn im Erkenner sogenannte "Cross-word"-Modelle verwendet werden. Bei diesen wird der Phonemkontext auch über Wortgrenzen hinweg berücksichtigt.

Einen konzeptionell entgegengesetzten Ansatz stellten Bahl et al. in [Bah91] vor. Die Autoren präsentierten ein binäres Entscheidungsbaumverfahren zur Gruppierung der Zustände von kontextabhängigen HMM-Modellen. Verfahren dieser Art sind auch unter der Bezeichnung CART (engl.: "Classification And Regression Trees") geläufig. Im Prinzip handelt es

sich bei dem vorgeschlagenen Algorithmus um ein Top-Down Clusterverfahren, bei dem die möglichen, binären Teilungen eines Baumknotens durch weitergehende phonetische Informationen vorgegeben werden.

4.1.2 Phonetische Entscheidungsbäume

Ausgangspunkt dieses Top-Down Clusterverfahrens ist eine einzelne, globale Gruppe, die alle vorgekommenen Lautkontexte einschließt. Diese wird iterativ aufgeteilt, wobei je Iteration eine Aufteilung erfolgt. Prinzipiell wären bei derartigen Verfahren beliebige Aufteilungen möglich. Im Rahmen sogenannter "phonetischer" Entscheidungsbäume werden die möglichen Teilungen einer Gruppe jedoch durch eine Reihe von phonetischen Einteilungen oder Eigenschaften vorgegeben. Diese können umfassen:

- einzelne Phoneme
- Phonemgruppen

Im ersten Fall besteht eine Gruppe nur aus einem Einzelelement. Meist werden alle verwendeten Phoneme (ca. 40-50) als Einzelgruppen aufgenommen [Bah91, Beu99]. Ergänzt werden sie durch komplexere Lautklassen, die phonetisch oder artikulatorisch [Sch89] begründet sein können. Einige Beispiele hierfür sind in Tabelle 4.1 wiedergegeben.

Gruppenbezeichnung	enth. Phoneme
Frikativ	/C/, /S/, /f/, /j/, /s/, /v/, /x/, /z/
Plosiv	/b/, /d/, /g/, /k/, /p/, /t/
Labial	/b/, /p/, /m/

Tab. 4.1: Beispiele für Phonemgruppen (in Sampa Notation).

Eine Einteilung könnte also beispielsweise lauten:

Teile in eine Gruppe, deren linker Kontext ein /a/ ist,
und eine Gruppe die links kein /a/ hat (entsprechende Anti-Menge).

Um die beste Einteilung in einem Iterationsschritt zu finden, werden *alle* vorgegebenen Einteilungen getestet. Diese Tests werden in diesem Zusammenhang auch mit dem Begriff "Fragen" (Ist links ein...?) bezeichnet. Die Bedeutung einer Frage wird anhand eines Gütekriteriums bewertet. Letztendlich wird eine Gruppe, d.h. ein Knoten des Entscheidungsbaums, anhand der Frage aufgeteilt, die die beste Bewertung erzielt. In der Literatur wurden verschiedene Gütemaße vorgeschlagen [Rog97, Duc97, Cho99], wobei jedoch die folgenden 2 Kriterien von primärer Bedeutung sind:

- Entropie
- Likelihood Differenz

Eine vergleichende Untersuchung beider Maße wurde von Rogina in [Rog97] vorgestellt. Es zeigte sich, dass bei gleichen Daten die Anwendung des Likelihood-Kriteriums zu einer besseren Performanz bezüglich der Worterkennungsrate führt. Grundlage beider Maße ist eine statistische Repräsentation der verfügbaren Daten. Aus Gründen des Rechenzeitaufwands wird meist ein einfaches unimodales Gaussmodell angewandt, d.h. die gesamten, einer Gruppe zugeordneten Daten werden durch eine einzige Normalverteilung $\mathcal{N}$ repräsentiert. Bei einer Teilung müssen für die entstandenen neuen Gruppen die Parameter $\boldsymbol{\mu}_{i1}, \boldsymbol{\Sigma}_{i1}$ bzw. $\boldsymbol{\mu}_{i2}, \boldsymbol{\Sigma}_{i2}$ neu bestimmt werden. Bei einer unimodalen Verteilung gestaltet sich die Neuschätzung der Parameter Mittelpunkt und Kovarianz (meist diagonal) entsprechend einfach:

$$\boldsymbol{\mu}_i = \frac{1}{N_P^i} \sum_{j=1}^{N_P^i} \boldsymbol{x}_j \tag{4.1}$$

$$\boldsymbol{\Sigma}_i = \frac{1}{N_P^i} \sum_{j=1}^{N_P^i} (\boldsymbol{x}_j - \boldsymbol{\mu}_i)(\boldsymbol{x}_j - \boldsymbol{\mu}_i)^T = \frac{1}{N_P^i} \sum_{j=1}^{N_P^i} \boldsymbol{x}_j \boldsymbol{x}_j^T - \boldsymbol{\mu}_i \boldsymbol{\mu}_i^T \tag{4.2}$$

Bei Verwendung einer multimodalen Verteilung muss bei der Parameterneubestimmung auf ein EM-Verfahren zurückgegriffen werden, was verglichen mit Gl. 4.1 bzw. 4.2 einen deutlichen Mehraufwand an Rechenzeit bedeutet. Unter Verwendung der unimodalen Repräsentation lässt sich die Likelihood einer Datenmenge angeben zu:

$$L(\boldsymbol{\mathcal{X}}_i) = \sum_{j=1}^{N_P^i} L(\boldsymbol{x}_j) = \sum_{j=1}^{N_P^i} \log p(\boldsymbol{x}_j | i) = \sum_{j=1}^{N_P^i} \log \mathcal{N}(\boldsymbol{x}_j, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \tag{4.3}$$

Die Summation in obiger Gleichung erstreckt sich über alle Mustervektoren N_P^i einer Gruppe i. Bei einer Teilung entstehen die beiden Kindknoten, hier bezeichnet mit $i1$ und $i2$. Um deren Likelihood zu bestimmen werden zuerst die Parameter der jeweiligen Normalverteilung neu berechnet, daran anschließend wird mit dieser neuen Verteilung die Likelihood gemäß Gleichung 4.3 bestimmt. Die Likelihood lässt sich nach Auflösung von Gl. 4.3 auch direkt aus den Varianzparametern der Gaussverteilung ableiten [Beu99]. Damit lässt sich der Likelihood-Gewinn [You94] durch die Teilung eines Knotens angeben zu:

$$\Delta L_i^q = L_{i1}^q(\boldsymbol{\mathcal{X}}_{i1}) + L_{i2}^q(\boldsymbol{\mathcal{X}}_{i2}) - L_i(\boldsymbol{\mathcal{X}}_i) \tag{4.4}$$

wobei $\{\boldsymbol{\mathcal{X}}_i\} = \{\boldsymbol{\mathcal{X}}_{i1}\} \cup \{\boldsymbol{\mathcal{X}}_{i2}\}$. Diese Likelihood-Differenz wird in einem Iterationsschritt t für alle zu diesem Zeitpunkt und für diesen Knoten möglichen Fragen berechnet. Die Auswahl, anhand welcher Frage q dieser Knoten in dieser Iteration zu teilen ist, wird anhand des maximalen Gewinns festgelegt:

$$q^* = \arg\max_q \Delta L_i^q \tag{4.5}$$

Ein mögliches Abbruchkriterium des Verfahrens ergibt sich direkt aus der Berechnungsgleichung 4.4 des Likelihoodgewinns. Der Gewinn aus der Spaltung eines Knotens sollte positiv sein. Etwas allgemeiner lässt die Abbruchbedingung formulieren zu:

$$\Delta L_i^{q^*} < L_C \tag{4.6}$$

wobei $L_C > 0$. Eine ergänzende Abbruchbedingung ergibt sich durch die Zahl der Datenvektoren in einem potentiellen Kindknoten. Fällt diese unter einen Grenzwert N_P^{min}, der für eine robuste Parameterschätzung nötig ist, so wird die Teilung dieses Knotens verworfen. Die globale Abbruchbedingung des gesamten Algorithmus ergibt sich also daraus, dass kein Terminalknoten mehr gefunden werden kann, dessen beide Kindknoten mehr als N_P^{min} Datenvektoren zur Verfügung haben. Der folgende Ablaufplan zeigt den gesamten Algorithmus.

```
Berechne globalen Mittelpunkt/Varianz (Wurzelknoten)
Wiederhole solange teilbare Blattknoten (Abbruchbed.) vorhanden
        Für Blattknoten:
                Berechne Likelihood-Gewinn für alle Fragen
                Teile anhand Frage mit maximalem Gewinn,
                        falls Abbruchbedingung nicht erfüllt
```

In der ursprünglichen, von Bahl et al. [Bah91] vorgeschlagenen Form wird der Algorithmus individuell für die Einzelzustände der kontextabhängigen Hidden-Markov-Modelle angewandt. In dieser Form werden für eine HMM-Struktur mit drei Zuständen insgesamt drei Entscheidungsbäume erzeugt. Die Zustände $i = 1..3$ aller Lautmodelle werden durch jeweils einen Baum zu Gruppen zusammengefasst. Weitergehende Ansätze lösen die Baum-zu-Phonem bzw. die Baum-zu-Zustand Zuordnung komplett auf, indem sie diese Information als Kontextfrage [Laz96, Pau97] in den Entscheidungsprozess integrieren. In diesen Entscheidungsbäumen tauchen also Fragen nach dem Phonem bzw. dem Zustand auf. Die Extremform dieses Vorgehens stellt ein globaler Entscheidungsbaum für alle Modelle dar. Laut Lazarides [Laz96] zeigen phonemspezifische Entscheidungsbäume jedoch keine nennenswerte Verbesserung gegenüber zustandsspezifischen Entscheidungsbäumen, insbesondere da in ersteren die Fragen nach dem Zustand sehr nahe am Wurzelknoten auftreten. Im folgenden erfolgt daher eine Betrachtung des Bahl'schen Originalansatzes.

Ausgehend von einem Basisphonemmodell, wird für jeden Zustand dieses Modells ein eigener Entscheidungsbaum aufgebaut. Bei der Generierung von Triphonmodellen sind bei beispielsweise $N_{Ph} = 50$ Lauteinheiten insgesamt $N_{Ph}^2 = 2500$ unterschiedliche Triphonmodelle je Phonem möglich. Werden weiterhin 3 Zustände je Modell angenommen, so ergibt sich ein gesamte Zustandszahl von $3 * 2500 = 7500$ je Basislaut. Bei der Anwendung des Entscheidungsbaumverfahrens auf Zustandsebene, werden jeweils 2500 Zustände mit individuellen Bäumen zu Gruppen zusammengefasst (s. Abb. 4.2). Je nach verfügbarer Trainingsdatenmenge, wird die Zahl der nötigen Codebücher (=Blattknoten) von 2500 auf 1..200 reduziert.

Der wichtigste Vorteil phonetischer Entscheidungsbäume ist in ihrer abbildenden Eigenschaft zu sehen. Aufgrund der Top-Down Struktur, die zu Beginn *alle* betrachteten Lautkombinationen einschließt und der ausschließlichen Teilung anhand phonetischer Gruppen ist eine fehlende Triphon-zu-Triphongruppe Zuordnung ausgeschlossen, d.h. auch wenn eine Lautkombination nicht im Trainingskorpus gesehen wurde, so kann anhand der phonetischen Entscheidungen innerhalb des Baums dennoch *eindeutig* eine Zuordnung zu einem Blattknoten und damit sogar zu einem repräsentativen, weil i.d.R. phonetisch ähnlichen, Modell erfolgen.

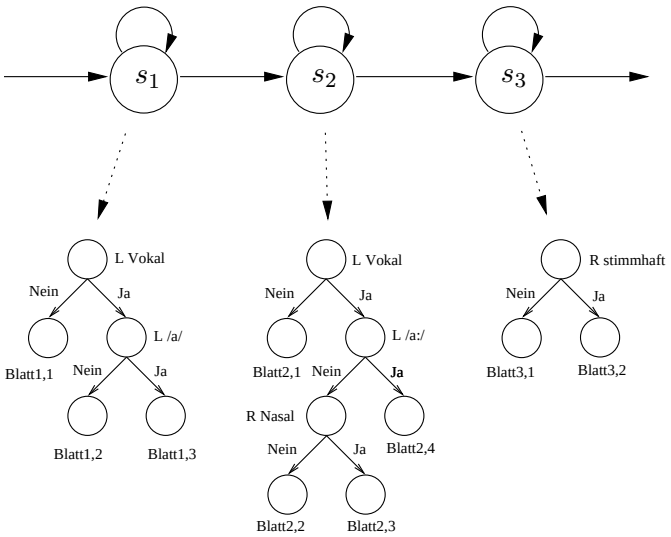

Abb. 4.2: Beispielbäume für 3 Zustände eines Basisphonems (L: linker Kontext, R: rechter Kontext).

Bei anderen Verfahren, wie beispielsweise Bottom-Up Gruppierung [ChC97, Fra00, Imp00], muss diese Eigenschaft erst durch eine geeignete "Fall-Back"-Strategie ausgeglichen werden.

4.1.3 Vom Entscheidungsbaum zum Erkenner-Modell

Wie in Abschnitt 1.3 erläutert, werden im verwendeten Spracherkennungssystem stochastische Hidden-Markov-Modelle zur statistischen Feinmodellierung sprachlicher Einheiten verwendet. Um eine hinreichende Genauigkeit bei der Repräsentation der Merkmalsverteilung eines Lauts zu erreichen, müssen hier sehr viele Gaussverteilungen je HMM eingesetzt werden. Die in dieser Arbeit untersuchten und trainierten Triphonsysteme arbeiten alle mit insgesamt ca. 35000 Gauss'schen Prototypen (s. Tab. 5.1). Für das Training dieser Verteilungen wurde im Rahmen dieser Arbeit das Segmental K-Means Verfahren implementiert (s. Kap. 2.3.1).

An dieser Stelle muss zwischen den finalen Modellen, die letztendlich im Erkennungssystem zum Einsatz kommen, und der reduzierten Modellierung, die im Entscheidungsbaumverfahren angewandt wird (s. Gl 4.3) unterschieden werden. Beide repräsentieren die selben phonetischen Einheiten, unterscheiden sich jedoch stark in der Zahl ihrer Parameter. Für die Generierung des Zustands-Tyings, d.h. im Entscheidungsbaumalgorithmus, werden i.d.R. unimodale Gaussmodelle vorgezogen [Noc97, Beu99]. Diese sind jedoch für die akustische Feinmodellierung im Erkennungssystem noch nicht optimal geeignet, da die Annahme der Unimodalität für die Feinmodellierung von sprachlichen Äußerungen zu grob ist. Um zu einem für den Erkenner passenderen Modell zu gelangen, ist es zweckmäßig, von einem unimodalen zu einem multimodalen Gauss'schen Mixturmodell überzugehen. Hierzu können die in Abschnitt 2.2 beschriebenen Clusterverfahren eingesetzt werden.

Speziell bei Triphonmodellen zeigte keines der beschriebenen Clusterverfahren nennenswerte Vor- oder Nachteile bzgl. der Erkennungsrate. Tabelle 4.2 zeigt einige Erkennungsergeb-

nisse von Modellen mit unterschiedlicher Initialisierung. Die Codebücher der Triphonmodelle wurden mittels des Segmental K-Means-Verfahrens trainiert und mit dem EMR-, K-Means, sowie LBG-Ansatz initialisiert.

Initialisierung	WER [%]
Emission-Ratio	25.4
K-Means Alg.	25.6
LBG-Alg.	25.2

Tab. 4.2: Vergleich der Worterkennungsrate von Triphonmodellen bei gleicher Parameterzahl jedoch unterschiedlicher Initialisierung.

Die Tabelle 4.2 macht deutlich, dass bei Triphonmodellen die Art der Initialisierung von eher untergeordneter Bedeutung ist. Dieses Ergebnis steht im Gegensatz zu den Ergebnissen bei der Initialisierung von Monophonmodellen [Fal99]. Hintergrund dürfte hierbei sein, dass bei Triphonen auf jeden Zustand, d.h. jedes gemeinsam verwendete Codebuch, i.d.R. deutlich weniger Mixturen entfallen als bei einem äquivalenten Monophonmodell. So liegt die Zahl der zustandsweisen Codebücher bei Monophonen bei $\approx$150 ($\hat{=}$ Zahl der HMM-Zustände), bei Triphonen dagegen bei ≈ 2500 (Gesamtzahl der gemeinsam verwendeten Codebücher). Bei gleicher Gesamtzahl an Verteilungen hat jedes Codebuch um etwa einen Faktor 10 weniger Dichten. Absolut sind es meist weniger als 20.

Der Kernpunkt der folgenden Experimente ist die Untersuchung der strukturellen Abhängigkeit zwischen Zustandsgruppierung und "externen" Einflussgrößen, wie hier die Sprechgeschwindigkeit (SR). Aus diesem Grund wird ausschließlich die individuelle Modellierung der Zustandsgruppen (1 Codebuch/Zustandsgruppe) betrachtet - andere Tyingmechanismen, z.B. auf Verteilungsebene [Hwa92], werden nicht eingesetzt.

4.2 Allgemeine Untersuchungen

4.2.1 Sprechgeschwindigkeit und Sprachmodell

Die Dekodierung einer sprachlichen Äußerung erfolgt durch Maximierung der als 1.1 bzw. 1.2 gegebenen Gleichung bzgl. aller möglichen Wortfolgen bei gegebener Musterfolge $\boldsymbol{\mathcal{X}}$. Bei der Suche nach der optimalen Wortfolge gemäß Gl. 1.2 bleibt die Apriori-Wahrscheinlichkeit $p(\boldsymbol{\mathcal{X}})$ der Musterfolge unberücksichtigt, da sie keine Auswirkung auf die Maximumsuche hat. Bei der eigentlichen Abarbeitung des Suchraums werden nur das akustische Modell $p(\boldsymbol{\mathcal{X}}|\boldsymbol{w})$, sowie das Sprachmodell $p(\boldsymbol{w})$ eingerechnet. Bei den Likelihood-Funktionen der akustischen Modelle handelt es sich jedoch um Wahrscheinlichkeits*dichte*funktionen, die numerisch einen vollkommen anderen Wertebereich einnehmen als die wortdiskreten Wahrscheinlichkeiten des Sprachmodells. Um dieses Defizit auszugleichen wird bei aktuellen Spracherkennungssystemen eine Gewichtung λ_{LM} des Sprachmodells eingesetzt. Die Maximierung erfolgt aus numerischen Gründen im logarithmischen Bereich. Aus Gl. 1.2 wird daher

	WER [%], SR-Klasse der Testdaten					
λ_{LM}	gesamt	s. langsam	langsam	mittel	schnell	s. schnell
9.0	27.4	23.2	30.0	26.3	*26.7*	*30.6*
10.0	26.9	22.5	29.6	25.5	*26.7*	31.4
11.0	26.7	22.9	28.9	25.2	27.3	30.9
12.0	26.1	21.9	28.2	24.3	27.1	31.4
13.0	25.8	22.3	28.2	*23.3*	27.0	32.3
14.0	26.4	21.9	28.1	24.3	28.0	33.6
15.0	26.7	*21.5*	*27.6*	24.6	28.6	35.0
16.0	27.5	22.5	28.1	25.1	30.7	36.3
17.0	28.1	22.9	28.1	26.0	30.9	37.2

Tab. 4.3: WER bei unterschiedlicher Gewichtung λ_{LM} des Sprachmodells.

$$\hat{\boldsymbol{w}} = \arg\max_{\mathbf{w}} p(\boldsymbol{\mathcal{X}}|\boldsymbol{w})p(\boldsymbol{w}) = \arg\max_{\mathbf{w}} \left(\log p(\boldsymbol{\mathcal{X}}|\boldsymbol{w}) + \log p(\boldsymbol{w})\right). \tag{4.7}$$

Unter Einbeziehung des Sprachmodellfaktors λ_{LM} wird hieraus

$$\hat{\boldsymbol{w}} = \arg\max_{\mathbf{w}} \left(\log p(\boldsymbol{\mathcal{X}}|\boldsymbol{w}) + \lambda_{LM} \log p(\boldsymbol{w})\right). \tag{4.8}$$

Die Optimierung des Sprachmodellfaktors λ_{LM} erfolgt üblicherweise empirisch unter Verwendung von Crossvalidierungsdaten. Die Mehrzahl der sprachlichen Äußerungen erfolgt jedoch bei einer mittleren Sprechgeschwindigkeit von ca. 12-14 Phonemen/s (vgl. Histogramm in Abb. 4.4). Das anhand des Sprachmodellfaktors λ_{LM} eingestellte Gleichgewicht wird also primär für mittlere Sprechrate erreicht, da für diesen Bereich die meisten Trainingsmuster vorliegen. Ein Abweichen von der mittleren Sprechrate führt also zu einer Über- bzw. Unterbewertung des Sprachmodells gegenüber dem akustischen Score. Eine Sprechgeschwindigkeitsanpassung des LM-Gewichts wurde auch von Martinez et al. in [Mar97] vorgeschlagen. Allerdings gehen die Autoren nicht näher darauf ein in welcher Form die Gewichtsoptimierung durchgeführt wurde, die zu den angegebenen Verbesserungen geführt hat.

Betrachtet man die Erkennungsergebnisse in Abhängigkeit von der Gesamtzahl der Blattknoten N_B in Tabelle 4.3 jedoch näher, so zeigt sich allerdings, dass sich hier im Prinzip zwei gegenläufige Tendenzen überlagern. In Abschnitt 3.4.1 konnte gezeigt werden, dass der akustische Score, d.h. die Likelihood des akustischen Modells $p(\boldsymbol{\mathcal{X}}|\boldsymbol{w})$ hochgradig durch die vorliegende Sprechgeschwindigkeit beeinflusst wird (s. Abb. 3.11 bzw. 3.15). Mit steigender Sprechgeschwindigkeit sinkt jedoch auch die mittlere Phonemdauer [Mar97] (s. auch Abschnitt 3.2.4). Damit sinkt auch die mittlere Wortlänge in Frames. Dies wiederum führt unmittelbar zu einem absolut höheren Gesamtscore eines Worts bei schneller Sprache, da insgesamt weniger (negative) Framescores je Wort aufakkumuliert werden. Setzt man dies in Relation zum Sprachmodell bei gegebenem Sprachmodellgewicht, so ist in diesem Fall der Einfluss des Sprachmodells deutlich höher.

SR-Klasse	λ_{LM}	Ausl. [%]	Einf. [%]	WER [%]	Ausl., rel. [%]	Einf., rel. [%]
s. schnell	9.0	4.1	5.4	30.6	13.4	17.6
	13.0	6.0	4.8	32.3	18.6	14.9
	17.0	7.6	5.1	37.2	20.4	13.7
mittel	9.0	1.8	8.4	26.3	6.8	31.9
	13.0	2.4	6.2	23.3	10.3	26.6
	17.0	3.8	5.1	26.0	14.6	19.6

Tab. 4.4: Auslassungen/Einfügungen als absoluter Fehler, sowie bezogen auf die Gesamtfehlerrate - ausgewertet für die mittleren und sehr schnelle Testsätze.

Die Addition des Sprachmodellanteils in Gl. 4.8 kann aufgrund des negativen (wegen $\log p(\boldsymbol{w})$) Beitrags gewissermaßen als "Bestrafungsterm" für einen Wortübergang interpretiert werden. Ein Höhergewichtung mit λ_{LM} erschwert demnach einen Wortübergang und führt indirekt zu einer Verringerung der Einfügungsfehler (s. Tab. 4.4). Eine analoge Auswirkung hat die durch erhöhte Sprechgeschwindigkeit induzierte Reduktion der Wortlänge. Durch die betragsmäßig verringerten Wortscores steigt im Verhältnis dazu die Bestrafung durch einen Wortübergang - die Zahl der Einfügungen sinkt ebenfalls.

4.2.2 Sprechgeschwindigkeit und Baumgröße

Vergleichende Erkennungsexperimente (s. Tab. 4.5) zwischen Mono- und Triphonmodellen zeigen, dass erstere auf sehr langsamer Sprache vergleichbare, bisweilen sogar bessere Erkennungsergebnisse als äquivalente Triphonstrukturen aufweisen. Wenngleich die Triphonmodellierung bei sehr langsamer Sprache noch keinen Vorsprung der Worterkennungsrate zeigt, so steigt mit zunehmender Sprechgeschwindigkeit der Performanzgewinn jedoch auch gegenüber den Monophonmodellen mit mehr Verteilungen kontinuierlich an. Bei mittlerer Sprechrate beträgt er bereits ca. $3-4\%$ absolut und wächst bei sehr schneller Sprache auf nahezu $7-10\%$ absolut an. Da die Monophonmodellierung als der Spezialfall des globalen Tyings der Triphonmodellierung betrachtet werden kann, stellt sich hierbei die Frage, inwieweit der *Verzweigungsgrad* eines Entscheidungsbaums Einfluss auf die Worterkennungsrate bei verschiedenen Sprechgeschwindigkeiten haben könnte.

			WER [%]				
Modell	N_B	N_{Bf}	sehr langsam	langsam	mittel	schnell	sehr schnell
Monophon	135	8448	21.9	32.2	29.1	34.5	41.0
Monophon	135	50000	20.0	31.3	28.5	31.5	38.2
Triphon	3506	35884	22.3	28.1	23.5	27.1	31.5

Tab. 4.5: Vergleich der WER von Monophon- und Triphonmodellen für verschiedene Sprechgeschwindigkeiten des Testkorpus.

Zur Veranschaulichung dieser Vermutung wurden ausgehend von der Triphongruppierung aus Tabelle 4.5 manuell Bäume unterschiedlicher Größe generiert. Durch Erhöhung der

Likelihood-Gewinnschwelle L_C (Gl. 4.6) kann der jeweilige Basisbaum beschnitten ('Pruning', engl. für: 'einen Baum beschneiden') werden, indem schrittweise diejenigen Terminalentscheidungen zusammengefasst werden, deren Gewinn unter die erhöhte Schwelle L_C fällt.

			WER [%]					
N_B	L_C	N_{Bf}	Total	sehr langsam	langsam	mittel	schnell	sehr schnell
3506	300	35884	25.8	22.3	28.1	23.5	27.1	31.5
2291	1000	36458	25.2	19.0	26.9	24.0	25.4	32.3
509	10000	34776	28.2	19.4	27.1	26.8	30.8	40.6

Tab. 4.6: WER für verschiedene Gesamtgrößen N_B, aufgeschlüsselt nach der Sprechgeschwindigkeit bei unverändertem LM-Gewicht $\lambda_{LM} = 13.0$ (optimiert für $N_B = 3506$).

	WER [%]					
N_B	gesamt	sehr langsam	langsam	mittel	schnell	sehr schnell
3506	25.8	21.5	27.6	23.3	26.7	30.6
2291	25.2	19.0	26.9	23.9	25.4	30.9
509	27.3	17.3	27.9	25.9	28.0	34.5

Tab. 4.7: WER für verschiedene Gesamtgrößen N_B, aufgeschlüsselt nach der Sprechgeschwindigkeit bei kategorieweise optimiertem LM-Gewicht λ_{LM}.

Auffallend in den Tabellen 4.6 bzw. 4.7 ist, dass die Reduktion der Blattknotenzahl zu keiner nennenswerten Verschlechterung der WER bei langsamer Sprechweise führt. Bei einer geeigneten Optimierung von λ_{LM} (s. Tab. 4.7) lässt sich bei langsamer Sprache sogar eine deutliche Reduktion der WER erzielen. Im Gegensatz hierzu steigt bei sehr schneller Sprache die Fehlerrate um über 9% an.

4.3 Bildung von Modellgruppen bezüglich der Sprechgeschwindigkeit

4.3.1 Ansatzpunkte zur klassenweisen Einteilung

Das Konzept eines Trainings sprechgeschwindigkeitsspezifischer Modelle, wurde - parallel zu dieser Arbeit - auch von anderen Autoren aufgegriffen [Pfa98b, Pfa00b, Zhe00]. Die Ansätze konzentrieren sich meist auf eine Einteilung der Trainingsdaten anhand definierter Geschwindigkeitskategorien und einem anschließenden individuellen Training mit den jeweiligen Daten.

Im folgenden werden verschiedene Kombinationsmöglichkeiten zur Generierung klassenspezifischer, kontextabhängiger Modelle untersucht und verglichen. Wie im vorigen Abschnitt beschrieben, erfolgt die Gruppierung von Triphoneinheiten vorzugsweise anhand phonetischer Entscheidungsbäume. Die Erzeugung der endgültigen Erkennermodelle läuft dabei in mehreren Stufen ab:

1. Zustandssegmentierung der Trainingsdaten
2. Zustandsweise Entscheidungsbäume mittels unimodaler Gaussmodelle
3. evtl. Optimierung der Entscheidungsbäume
4. Übergang zu Mixturverteilungen
5. Training der Mixturverteilungen

Die im Vorfeld aufgeführten Arbeiten konzentrierten sich zur Bildung der Klassenmodelle auf Punkt 5 obiger Aufzählung. Die Trennung anhand der Sprechgeschwindigkeitsklassen kann im Prinzip jedoch bereits in den Punkten 2-4 ansetzen. Die beiden Schritte 4 und 5 können bereits bei reinen Monophonmodellen zur klassenweisen Optimierung herangezogen werden. Pfau zeigte dies für den Trainingsaspekt in [Pfa98b]. Für die Initialisierung der Mixturverteilungen konnte dies im Rahmen dieser Arbeit in [Fal99] gezeigt werden. Die Kernidee des dort vorgestellten und in Abschnitt 2.2.3 beschriebenen Clusterverfahrens ist die Generierung einer für spezifische Sprechgeschwindigkeiten optimierten Anzahl von Normalverteilungen. Bei kontextabhängiger Modellierung kommen die Schritte 2 und 3 als mögliche Ansatzpunkte zusätzlich hinzu. Im folgenden werden die verschiedenen Ansatzpunkte und Kombinationsmöglichkeiten zur Berücksichtigung von Sprechgeschwindigkeitskategorien im Überblick aufgezeigt. Als Referenz sind die Indizes aus obiger Aufzählung angegeben.

1. (ROS-unabhängig: gemeinsamer Entscheidungsbaum, gemeinsames Training)
2. gem. Entscheidungsbaum (2-4), klassenspez. Training (5)
3. gem. Entscheidungsbaum(2), klassenspez. Pruning(3,4), gem. Training (5)
4. gem. Entscheidungsbaum(2), klassenspez. Pruning(3,4), klassenspez. Training (5)
5. Integration in Entscheidungsbaum (2)

Der erste Schritt, ohne Berücksichtigung der Sprechrate, ist aufgeführt, um die Ansatzpunkte zuordnen zu können. Punkte 3 und 4 stehen für 2 untersuchte Ansätze wie mittels Pruning die Performanz von Bäumen auf entsprechenden Daten beeinflusst werden kann. Eine Beschreibung des Vorgehens findet sich im Abschnitt 4.3.4. Der Ansatz, die Sprechgeschwindigkeitsklassen als Kontextinformation direkt im Entscheidungsprozess zu berücksichtigen, wird in Abschnitt 4.4 diskutiert. Abb. 4.3 gibt einen schematischen Überblick über die verschiedenen Vorgehensweisen.

4.3.1.1 Einteilung der Daten

Die unter den nachfolgenden Abschnitten beschriebenen Experimente bauen auf einer satzweisen Einteilung der Trainingsdaten auf. Für jeden Turn wurde individuell die mittlere Sprechgeschwindigkeit anhand der Phonemrate (Gl. 3.1) bestimmt, wobei der Beobachtungszeitraum T_{Beob} (Gl. 3.1) der Turndauer entspricht. Pausen und andere nicht-sprachliche Segmente wurden aus der Bewertung entfernt. Die sich ergebende Verteilung der Raten ist in

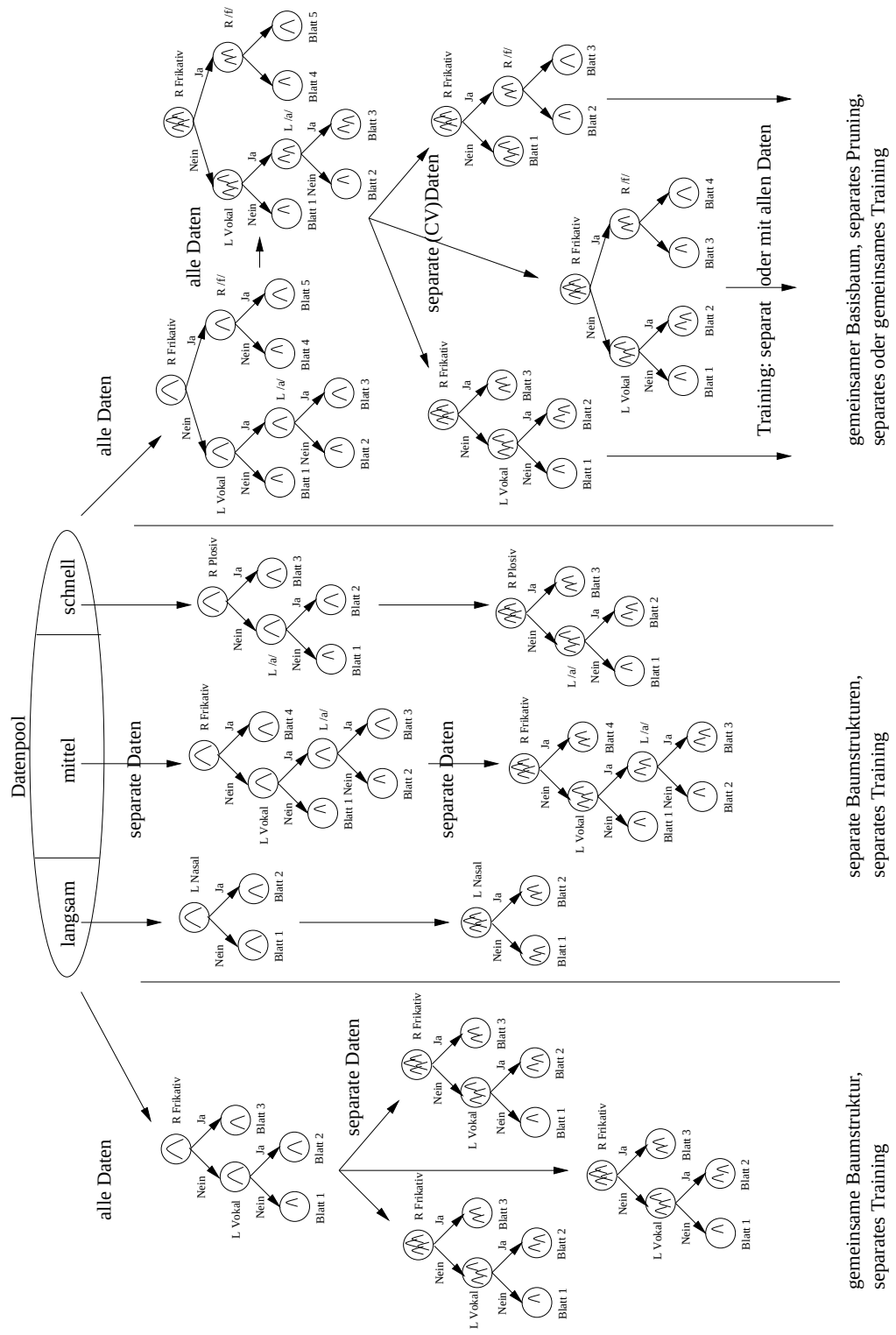

Abb. 4.3: Übersicht über verschiedene Vorgehensweisen zur Erzeugung sprechgeschwindigkeitsspezifischer Modellgruppen.

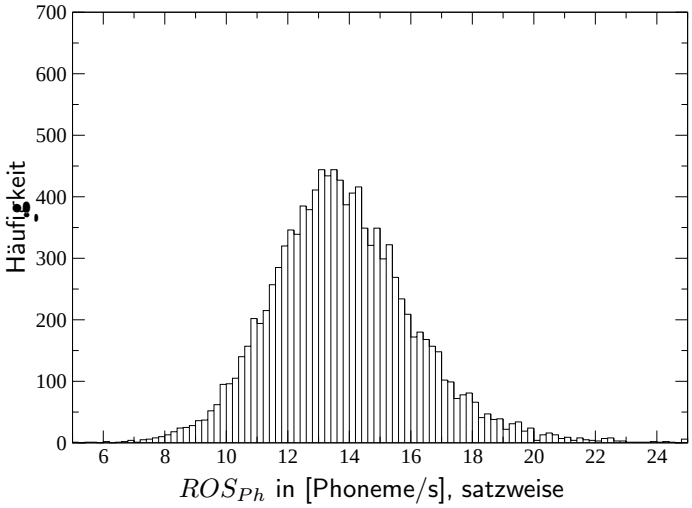

Abb. 4.4: Sprechgeschwindigkeitsverteilung der Trainingsäußerungen des Verbmobil-Korpus. ($\mu_{ROS} = 13.82[Phoneme/s], \sigma_{ROS} = 2.56[Phoneme/s]$)

Abb. 4.4 abgebildet.

Die mittlere Phonemrate beträgt $\mu_{ROS} = 13.82[Phoneme/s]$, bei einer Standardabweichung von $\sigma_{ROS} = 2.56[Phoneme/s]$. Die Einteilung der Turns wurde analog zu den Gleichungen 3.13 für drei Klassen, bzw. 3.14 für fünf Klassen durchgeführt. Prinzipiell stellt sich das Problem der Klasseneinteilung bei der Berücksichtigung von generellen Kontextmerkmalen, wie beispielsweise Sprechgeschwindigkeit, stets an zwei Stellen. Einerseits müssen, wie beschrieben, die Trainingsdaten in die definierten Klassen unterteilt werden. Darüber hinaus muss für die Erkennungsphase eine Selektion der zu verwendenden Modelle getroffen werden. Die Auswahl kann sowohl explizit, anhand eines Sprechgeschwindigkeitsmaßes, als auch implizit nach der Erkennung getroffen werden. Eine robuste und schnelle explizite Selektion kann mit dem in Abschnitt 3.3 vorgestellten Klassifikationssystem erfolgen. Bei impliziter Auswahl erfolgt die Selektion anhand des besten Gesamtscores aller Klassen. Eine frühzeitige, explizite Entscheidung für eine Modellklasse birgt das Risiko einer Fehlauswahl und damit einer u.U. starken Fehlerkennung. Die implizite Selektion anhand des erzielten Scores nach der Erkennung vermeidet dieses Risiko, allerdings unter der Prämisse alle Modelle bewerten zu müssen, was bei K Klassen den $K-$fachen Rechenaufwand bedeutet.

4.3.2 Gemeinsamer Entscheidungsbaum - Klassenweises Parametertraining

Wie am Anfang dieses Abschnitts bereits angeführt, gibt es verschiedene Ansätze, Sprechgeschwindigkeitsinformation in die Gruppierung bzw. Generierung von Triphonmodellen zu integrieren. Der Naheliegendste besteht in einem separaten Training der HMMs. Der Ansatz wurde, parallel zu dieser Arbeit, von Zheng [Zhe00], sowie, äquivalent für Monophonmodelle, von Pfau [Pfa98b, Pfa00b] untersucht. Ausgangspunkt ist hierbei ein gemeinsamer Entscheidungsbaum für jeden Zustand, der mit dem gesamten Trainingsmaterial (des jeweiligen Zustandes) gezüchtet wird. Danach wird das Datenmaterial, wie bereits beschrieben, in Sprechgeschwindigkeitskategorien unterteilt. Für jede der Klassen werden mit den entspre-

chenden Daten individuelle Mixturverteilungen trainiert.

Pfau verglich in [Pfa98b, Pfa00b] das Training mittels ML und MAP (s. Kap. 2.3.2 und 2.3.3) zur Generierung von spezialisierten Monophonmodellen für schnelle Sprache. Sowohl die ML- als auch die MAP-geschätzten, spezialisierten Modelle zeigen eine reduzierte Wortfehlerrate auf schneller Sprache, wobei sich der MAP-Ansatz (2% absolute Verbesserung) im Vergleich zu ML (0.9%) als besser geeignet erweist. Ein ebenfalls auf drei Geschwindigkeitsklassen basierender Ansatz zur statistischen Repräsentation der sprechgeschwindigkeitsabhängigen Aussprache wurde von Zheng et al. in [Zhe00] vorgestellt. Die Autoren verwendeten, analog zu Pfau, "Bayesian Smoothing" um die spezifischen HMM-Parameter abzuleiten. Zheng et al. setzten in ihrem System jedoch kontextabhängige Einheiten zur Lautmodellierung ein. In beiden Fällen wurde die Ableitung der sprechgeschwindigkeitsspezifischen Modelle lediglich durch Neuschätzung der Verteilungsparameter durchgeführt. Die Struktur der Kontextmodellierung blieb unberücksichtigt.

		WER [%] (Sprechgeschw.Kategorie)				
Train.Alg	Modellklasse	sehr l.	langsam	mittel	schnell	sehr s.
ML	langsam	25.3	28.8	25.6	29.4	36.8
	mittel	22.5	28.8	24.7	30.5	35.3
	schnell	28.4	34,2	28.8	30.7	33.4
Basismodelle		22.3	28.1	23.5	27.1	31.5

Tab. 4.8: WER für kategorial, ML-trainierte Modelle bei unterschiedlichen Sprechraten in den Testdaten.

Tabelle 4.8 zeigt deutlich die Nachteile eines kategorieweisen ML-Trainings, basierend auf einer vollständigen Teilung des Trainingsmaterials. Bei allen Modellkategorien ergibt sich für nahezu alle Kategorien des Testmaterials eine deutliche Verschlechterung der Erkennungsleistung. Der Grund für diese Verschlechterung dürfte in der bereits erwähnten Ausdünnung des Trainingsmaterials liegen. Bei dem eingesetzten ML-Training kommt dieser Effekt besonders stark zur Geltung. Pfau beschrieb ähnliche Ergebnisse mit Monophonmodellen. Im Vergleich zu der von Pfau berichteten Verschlechterung, fällt sie bei Triphonmodellen insbesondere für 'schnell' und 'langsam' deutlich stärker aus. Für das Training dieser Klassen scheint ein ML-Training ohne stärkeres Parametertying [Hwa92, Wre01] nicht geeignet. Aus diesem Grund wurden in dieser Arbeit weitere Trainingsverfahren auf ihre Verwendbarkeit untersucht: zum einen das auch von Pfau vorgeschlagene Maximum Aposteriori-Training (MAP, [Pfa00b, Gau92]) und zum anderen das robuste Maximum-Likelihood-Linear-Regression (MLLR,[Leg95]) Training. Eine Diskussion dieser Verfahren findet sich in Abschnitt 2.3.3 bzw. 2.4.

Um diese Verfahren effektiv einsetzen zu können, werden in einem ersten Schritt mit den gesamten Daten initiale, generische Erkennermodelle trainiert. Ausgehend von diesen Modellen lassen sich, mittels der kategorieweisen Daten und dem jeweiligen Trainingsverfahren, spezialisierte Modelle schätzen. Um eine Aussage über die Empfindlichkeit der Varianzparameter

zu erhalten, erfolgt das MAP-Experiment sowohl mit mit einer Adaption aller NV-Parameter, als auch - wie bei MLLR - mit einer ausschließlichen Anpassung der Gauss-Mittelpunkte.

		WER [%] (Sprechgeschw.Kategorie)				
Train.Alg	Modellklasse	sehr l.	langsam	mittel	schnell	sehr s.
Basismodelle		22.3	28.1	23.5	27.1	31.5
MAP	langsam	22.1	28.1	23.8	26.7	32.5
	mittel	22.1	28.1	23.7	26.8	32.3
	schnell	22.1	27.7	24.2	27.5	32.8
MAP,nur MP	langsam	22.7	28.8	24.7	28.2	34.4
	mittel	22.5	28.1	24.0	28.2	34.2
	schnell	22.7	28.0	24.9	27.2	32.6
MLLR, nur MP	langsam	21.9	28.5	23.6	27.0	33.0
	mittel	21.1	28.3	23.6	27.4	32.6
	schnell	21.7	28.4	23.6	26.8	33.6

Tab. 4.9: Wortfehlerraten der kategorial trainierten Modelle bei unterschiedlichen Sprechraten in den Testdaten.

Sowohl die MAP-, als auch die MLLR-trainierten Modelle zeigen eine, im Vergleich zum reinen ML-Nachtraining, deutlich verbesserte Performanz. Die Auswirkungen der durch die Teilung bedingten Datenabnahme können hierdurch effektiv reduziert werden. Auffallend in Tab. 4.9 ist jedoch, dass nur bei sehr wenigen Kategorien ohne zusätzliche Optimierung (z.B. LM) eine merkliche Verbesserung der WER im Vergleich zu den Basismodellen erzielt werden kann.

4.3.3 Klassenweise Entscheidungsbäume

Eine noch tiefere Integration der Sprechgeschwindigkeitsinformation wird erzielt, wenn der den Modellen zugrundeliegende Entscheidungsbaum individuell für jede Sprechgeschwindigkeitsklasse neu entfaltet wird. Hierfür wird das Sprachmaterial apriori in die gewählten Sprechgeschwindigkeitsklassen unterteilt. Die Unterteilung in die drei Klassen langsam, mittel und schnell erfolgte wiederum anhand der satzweisen Phonemrate. Damit kann für jede Klasse ein eigener Entscheidungsbaum generiert werden, wobei bei K Klassen entsprechend K unterschiedliche Gruppierungen für die Triphon HMM-Zustände entstehen. Nachteil ist hierbei, dass aufgrund des unterschiedlichen Tyings Trainingsverfahren (z.B. MAP), die von robusten Apriori-Verteilungsparametern eines Initialbaums ausgehen, nicht mehr unmittelbar eingesetzt werden können.

Tabelle 4.10 fasst die Ergebnisse zusammen. Wie im vorigen Abschnitt bereits erläutert, stellt das ML-Training ein Haupthindernis beim Einsatz individueller Entscheidungsbäume dar. Dies zeigt sich deutlich auch in den stark verschlechterten Erkennungsergebnissen der Tabelle 4.10. Die vollständige Datentrennung bereits während der Baumerzeugungsphase führt zu schlechteren Ergebnissen, als das alleinige, kategorieweise ML-Training der Mixturpara-

	Kategorie der Testsätze				
Modellklasse	sehr l.	langsam	mittel	schnell	sehr s.
langsam	22.5	28.4	27.0	30.7	39.6
mittel	25.3	27.6	26.3	30.9	33.9
schnell	24.8	33.9	28.1	33.6	34.1

Tab. 4.10: WER für ML-trainierte Modelle mit individuellen Entscheidungsbäumen.

meter (s. Tab. 4.8). Dies Ergebnis bestätigt jedoch, dass bei blattspezifischen Codebüchern die Struktur der Entscheidungsbäume maßgeblichen Einfluss auf die Erkennungsleistung hat.

4.3.4 Modellgenerierung durch Crossvalidierung

Eine Schwierigkeit bei der Optimierung der Baumstruktur besteht darin, die Abbruchparameter des Entscheidungsbaumalgorithmus sinnvoll einzustellen. In erster Linie handelt es sich dabei um die Parameter N_P^{min} und L_C. Bei den verwendeten Verbmobil Daten kann für N_P^{min} ein Wert von $N_P^{min} \approx 50 \ldots 250$ angesetzt werden. Die resultierende Baumgröße N_K hängt also letztendlich nicht allein von der Menge der Daten und ihren Eigenschaften ab, sondern maßgeblich auch von der Wahl der Parameter. Selbst die Variation innerhalb des angegebenen Spielraums für N_P^{min} hat bereits starke Auswirkungen auf N_K. Die Frage wie weit der Baum vergrößert werden kann, um die generalisierende Eigenschaft nicht zu verlieren, ist in dieser Form also stark von der Erfahrung des Systemdesigners abhängig.

Um diesem Schwachpunkt entgegenzutreten, wurden in der Literatur einige Vorschläge präsentiert. Als sinn- bzw. wirkungsvoll hat sich vor allem der Einsatz von Crossvalidierungsdaten (CV-Daten) [Noc97, Rog97] erwiesen. Die Idee ist, den entstehenden Entscheidungsbaum auf seine Generalisierungsfähigkeit zu prüfen, indem die Likelihood der CV-Daten ausgewertet wird. Je nach Ergebnis auf diesen Daten kann der Baum weiter vergrößert werden oder muss beschnitten werden. Bei der Anwendung dieses Verfahrens wird i.d.R. zuerst ein überzüchteter, d.h. sehr großer Baum generiert. Dieser wird, anhand der Ergebnisse auf den CV-Daten, solange beschnitten, als dadurch ein Gewinn auf den CV-Daten erzielbar ist.

Aufgrund der Ergebnisse in Tab. 4.6 ist es naheliegend, zu versuchen, die Entscheidungsbäume individuell für spezifische Sprechgeschwindigkeiten anzupassen. Um eine größenmäßige Anpassung der entstehenden Baumstrukturen zu erreichen, wird ein Baum nur auf einer Hälfte der Trainingsdaten gezüchtet. Die restlichen Trainingsdaten werden in Sprechratenkategorien unterteilt. Diese werden als CV-Daten verwendet, um den Ursprungsbaum zu beschneiden. Alle zugeordneten Codebücher der drei Bäume werden mittels Clusterverfahren initialisiert und erhalten die annähernd gleiche Anzahl an Basisfunktionen.

Die Modelle auf denen Tabelle 4.11 basiert, wurden alle mit dem gesamten Trainingsdaten trainiert (Segmental K-Means). Auffallend dabei ist, dass weniger das unterschiedliche Trainingsmaterial Auswirkungen zu haben scheint, als vielmehr die Anzahl entstehender Blattknoten. Offenbar scheinen, speziell für langsame Sprache, Bäume mit einer geringeren

	WER, SR Kategorie der Testdaten					
Modellklasse	sehr l.	langsam	mittel	schnell	sehr s.	N_B
langsam	21,5	24,5	23,2	28,9	31,5	2091
mittel	21,5	25,2	23,4	27,3	32,2	2487
schnell	*18,3*	24,5	24,2	28,9	32,3	1516
Basissystem	22,7	26,9	23,6	26,8	32,2	5223

Tab. 4.11: WER für unterschiedliche, ROS-spezifisch geprunte Bäume (kategorial, jedoch mit Geschlecht als zusätzlicher Kontextfrage).

Zahl von Blattknoten (bei blattspezifischen Codebüchern, d.h. kein Tying der Verteilungsparameter [Wre01, Hwa92]) eine bessere Performanz zu bieten als tendenziell größere Bäume.

4.3.5 Erkennernahe Crossvalidierung: Knotenbäume

Die im vorangegangenen beschriebenen Entscheidungsbäume haben den Nachteil, dass die erzeugten Bäume, d.h. deren Struktur und Größe, auf einem reduzierten Modell fußen. Während des eigentlichen Entscheidungsbaum-Algorithmus wird lediglich ein Modell mit einer Normalverteilung je Knoten verwendet. Speziell bei Knoten, die sich nahe am Wurzel-Knoten befinden, ist die Zahl der modellierten Vektoren noch sehr hoch - die Annahme einer unimodalen Verteilung ist daher u.U. nicht gerechtfertigt. Andererseits ist die Verwendung durchaus sinnvoll, da im Stadium des Entscheidungsbaums mehr die globale Lage von phonetischen Gruppen im Merkmalsraum modelliert werden soll, als deren explizite Ausprägung [Noc97]. Es kann also ein unimodales Modell verwendet werden, um die Grobstruktur zu finden. Im Anschluss daran kann mit mehr Verteilungen die Modellierung der Feinstruktur vorgenommen werden. Daher wurde eine Erweiterung des Pruning Verfahrens aus dem vorhergehenden Abschnitt entwickelt, welche von voll trainierten Erkennermodellen ausgeht.

Auf die Verwendung des Entscheidungsbaumverfahrens zur Verknüpfung von Triphonzuständen wurde bereits im Vorfeld eingegangen. Für die Erzeugung der Erkennermodelle sind insbesondere die Triphongruppen, die auf die Terminal(=Blatt)knoten entfallen, von Bedeutung. Durch Mixturerhöhung werden für diese Gruppen adäquate Verteilungen generiert, die im Erkennungsvorgang Verwendung finden. Die bauminternen Knoten (=Nicht-Blattknoten) sind in diesem Stadium aus modellierungstechnischer Sicht nicht mehr von Bedeutung. Diese Struktur sei im folgenden als “Blattbaum” bezeichnet. Nichtsdestoweniger stellen auch die bauminternen Knoten sinnvolle Gruppierungen von Triphonzuständen dar. Die Experimente mit Mono- und Triphonmodellen (s. Tab. 4.5, 4.6) haben gezeigt, dass eine Abhängigkeit zwischen Sprechgeschwindigkeit und Genauigkeit der Kontextmodellierung besteht. Aus diesem Grund wurde ein Ansatz untersucht, bei dem zusätzlich zu den Verteilungen in den Terminalknoten auch Verteilungen für die Zwischenknoten trainiert werden. Diese voll trainierte Struktur wird im folgenden als “Knotenbaum” bezeichnet.

4.3.5.1 Training eines Knotenbaums

Um für ein gegebenes Triphon die Zuordnung zur zu verwendenden WDF zu erhalten, muss der betrachtete Zustands-Blattbaum im Prinzip immer bis zum Erreichen eines terminalen Knotens durchlaufen werden. Die diesem Blattknoten zugeordneten Codebücher werden in der Erkennungsphase zur Berechnung der Emissionwahrscheinlichkeit herangezogen. Bei Knotenbäumen soll im Prinzip der komplette Baum modelliert werden. Jeder einzelne Knoten im Baum, der ja eine Gruppe von Triphonen darstellt, erhält einen individuellen Satz von Normalverteilungen. Kritischer Punkt ist die Gesamtzahl der Verteilungen die, unabhängig von der Schichttiefe, konstant sein sollte. Die Schichttiefe beschreibt die Anzahl der Knotenteilungsebenen, beginnend mit dem Wurzelknoten (s. Abb. 4.9). Erreicht wird dies dadurch, dass ein Vaterknoten k immer in etwa genauso viele Dichten enthält wie seine beiden Kinderknoten k_{ja} und k_{nein} zusammen, d.h.

$$N^{Bf}(k) \approx N^{Bf}(k_{ja}) + N^{Bf}(k_{nein}) \tag{4.9}$$

Ein Knotenbaum kann (u.a.) aus einem bestehenden Blattbaum rekonstruiert werden, in dem die Verteilungen in den Blattknoten als Ausgangspunkt verwendet werden. Es genügt die Codebücher der Blattknoten durch Mixturerhöhung zu initialisieren. Die initialen, intrinsischen Codebücher können durch fortgesetztes Zusammenfassen der Codebücher der jeweiligen Kindknoten erzeugt werden. Die Anzahl der Basisfunktionen wird somit kleiner, je weiter der Zustand vom Root(Wurzel)-Knoten entfernt ist. Abb. 4.5 veranschaulicht dieses Konzept.

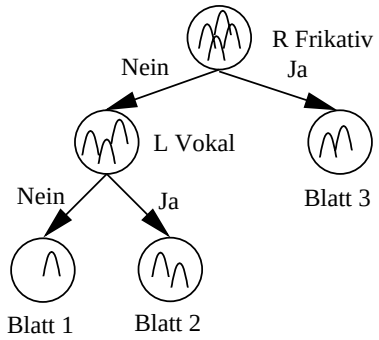

Abb. 4.5: Knotenbaum mit (Erkenner)WDF in jedem Knoten.

Ähnlich dem Training eines "normalen" Blattbaums kann auch das Training eines Knotenbaums erfolgen. Der dem Knotenbaum zugrundeliegende Blattbaum wird benutzt, um mittels Forced-Viterbi eine Segmentierung der Trainingsäußerungen zu erhalten. Im Unterschied zum reinen Blattbaum werden die durch die Segmentierung zugeordneten Trainingsvektoren jedoch benutzt, um die Codebücher *aller* Knoten nachzuschätzen, die durchlaufen werden müssen, um den jeweiligen terminalen Blattknoten zu erreichen. Abb. 4.6 zeigt dies schematisch.

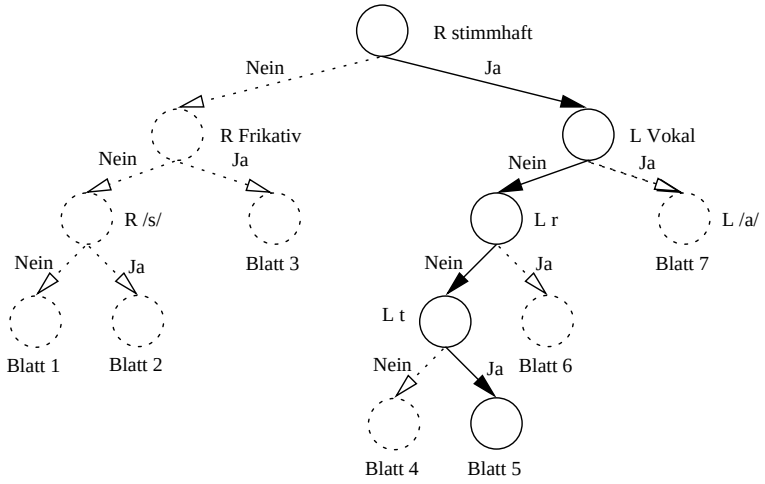

Abb. 4.6: Knoten entlang eines Pfads vom Wurzel zum Terminalknoten.

4.3.5.2 Knotenbaum in der Erkennung: Maximale Parsingtiefe und Abhängigkeit vom Sprachmodell

Um die Knotenbaumstruktur in der Erkennung einsetzen zu können, muss festgelegt sein, wie weit der Baum durchlaufen werden darf/muss, um die Triphon-zu-WDF Zuordnung zu finden. Wird der Baum immer bis zum jeweiligen terminalen Knoten durchlaufen, dann entspricht der Baum dem äquivalenten Blattbaum. Um die Codebücher der intrinsischen Knoten dynamisch ansprechen zu können, kann eine maximale Parsingtiefe (PT_{max}) für den Knotenbaum vorgegeben werden. Der Parameter PT_{max} begrenzt bei der Suche nach dem zugehörigen Codebuch die Zahl der erlaubten Knotenverzweigungen, die durchlaufen werden dürfen. Aufgrund der vorwärtsgerichteten Struktur des Entscheidungsbaums ist dies gleichbedeutend mit der Zahl der zugelassenen Schichten (s. Abb. 4.9). Durch die mögliche Variation des Parameters PT_{max} kann dieser Ansatz als ein dynamisches Pruning interpretiert werden. Die Motivation für diesen Ansatz ist in der in 4.2.2 getroffenen Vermutung zu suchen, dass für langsamere Sprache ein Baum mit geringerer Blattzahl vorteilhafter ist. Durch eine Verringerung des Parameters PT_{max} wird genau dies erreicht. Bei einem globalen Parameter, d.h. der Anwendung auf alle Phonemmodelle, werden die Basisbäume allerdings ungleichmäßig stark beschnitten. Tab. 4.12 zeigt dies beispielhaft für die Phoneme /n/ und /z/.

PT_{max}	2	4	6	∞
	#(Quasi)Blattknoten			
/n/	2	8	29	96
/z/	2	7	14	26

Tab. 4.12: Vergleich der Zahl der quasi-Blattknoten bei gegebener maximaler Parsingtiefe (jeweils Baum des mittleren Zustands s_2) für die Phoneme /n/ und /z/.

So zeigt ein Vergleich der Anzahl der entstehenden Quasi-Blattknoten, dass beim Phonem

/n/ bei weitem mehr Knoten abgeschnitten werden, als dies beispielsweise bei /z/ der Fall ist. Der Knotenbaum sollte allerdings nicht ohne Modifikation weiterer Erkennungsparameter eingesetzt werden. Speziell der Faktor λ_{LM}, mit dem das Sprachmodell gegenüber den akustischen Scores gewichtet wird, ist auf das Scoreniveau der Codebücher der terminalen Knoten optimiert. Bei einer Abweichung im akustischen Scoreniveau ergibt sich eine Verschiebung im Gleichgewicht zwischen Sprachmodell und akustischem Modell. Die Abweichung wird mit dadurch verursacht, dass Knoten, die näher am Wurzelknoten liegen, zwar einerseits durch mehr Normalverteilungen modelliert werden, andererseits deren Mixturkoeffizienten aber aufgrund der Normierungsbedingung entsprechend niedriger ausfallen. Der Language-Modell Faktor λ_{LM} muss daher zusätzlich zur Parsingtiefe PT_{max} verändert werden. Bei Verwendung der MFCC42-Vorverarbeitung, werden optimale Werte für $PT_{max} = 0$ und $LM = 8.0$ bzw. $PT_{max} = \infty$ und $LM = 13.0$ erreicht. In diesem Wertebereich $LM = 8..13.0$ muss λ_{LM} daher angepasst werden. Versuche mit verschiedenen Einstellungen von PT_{max} und λ_{LM} haben folgende Optima erzielt (s. Tab. 4.13):

Modellklasse	WER [%] (Basissystem)	WER [%] (optimal)	PT_{max}	λ_{LM}
sehr langsam	22.5	20.2	∞	13.0
langsam	24.8	23.7	8	13.0
mittel	22.0	21.8	∞	13.5
schnell	27.4	25.9	3	7.5
sehr schnell	32.2	30.7	8	11.0

Tab. 4.13: WER Optima für λ_{LM} und PT_{max} (geschlechtsabh. Modelle).

Die erzielten Werte weisen jedoch ein nicht konsistentes Verhalten auf. Prinzipiell sollte sich bei langsamer Sprache ein Optimum für kleine Werte von PT und λ_{LM} und bei schneller Sprache für hohe Werte von PT und λ_{LM} ergeben. Dies konnte in dieser Weise durch die Versuche noch nicht bestätigt werden. Ein Grund hierfür mag darin liegen, dass die Annahme einer globalen, phonemunabhängigen Parsingtiefe PT_{max} zu grob ist. Allerdings zeigt die Tabelle sehr deutlich, dass gegenüber dem Basissystem allein durch die Modellierung noch Potential zur Verbesserung steckt. Darüber hinaus wurde bei diesen Experimenten wiederum sichtbar, dass ein konsistenter Zusammenhang zwischen Sprachmodellfaktor und Sprechgeschwindigkeit besteht, wie Tab. 4.14 zeigt:

Modellklasse	WER [%] (Basissystem, $\lambda_m = 13$)	WER [%] (optimal)	λ_{LM}
sehr langsam	22.5	20.2	16.0
langsam	24.8	24.7	13.5
mittel	22.0	21.8	13.5
schnell	27.4	27.0	13.5
sehr schnell	31.1	30.7	12.0

Tab. 4.14: WER Optima für λ_{LM} bei $PT_{max} = \infty$ (geschlechtsabh. Modelle).

Wie bereits in Tab. 4.3 zeigt sich auch hier, dass für langsamere Sprache bei einem höherem Wert für λ_{LM} eine bessere Erkennungsleistung erzielt wird. Analog wird für schnelle Sprache bei einem niedrigeren λ_{LM}-Wert eine bessere Performanz erzielt. Die Ursachen dieser Abhängigkeit zwischen Sprachmodellfaktor und Sprechrate wurden bereits in Abschnitt 4.2.1 diskutiert.

Da eine globale Parsingtiefe noch sehr statisch ist, wird daher im nachfolgenden Abschnitt ein Pruningansatz vorgestellt, der eine verbesserte, wenngleich auch nicht mehr dynamische, Selektion der Terminalschicht erlaubt.

4.3.5.3 Reduktion vom Knotenbaum zum Blattbaum: Pruning

Ziel der Einführung von Knotenbäumen war eine genauere Modellierung der verfügbaren Trainingsdaten. Durch die Verwendung von Mixtur-Codebüchern in den Baumknoten ist diese erweiterte Struktur näher an den letztendlichen Erkennermodellen. Der erweiterte Baum ermöglicht nun eine genauere Analyse und Selektion von Knoten, die für die Repräsentation der Daten nötig sind. Durch klassenspezifisches Pruning, d.h. selektives, iteratives Abschneiden von Terminalknoten, wird der trainierte Knotenbaum wieder zu einem Blattbaum "reduziert". Für den eigentlichen Pruningvorgang bieten sich zwei Pruningstrategien an: Score- und Count-basiertes Pruning. Im folgenden wird der erstere Ansatz, der bereits für unimodale Verteilungen vorgestellt wurde, näher betrachtet.

Wie in Abschnitt 4.3.5.1 erläutert, können die jeweiligen Terminalknoten des Blattbaums zur Viterbi-Segmentierung der Crossvalidierungsdaten benutzt werden. Ein Mustervektor wird durch die Codebücher aller intrinsischen Knoten (s. Abb. 4.6), die zwischen dem, anhand der Viterbi-Segmentierung zugewiesenen Terminalknoten und dem Wurzelknoten liegen, bewertet. Der erzielte Score wird in jedem dieser Knoten aufakkumuliert. Somit könnte bereits für einen einzelnen Vektor entschieden werden, welches intrinsische Codebuch die beste Bewertung erzielt. In jedem Knoten wird stets die gleiche Anzahl von Mustervektoren bewertet, wie in den beiden unmittelbaren Ja/Nein-Kindknoten zusammen. Daher kann die Entscheidung, ob eine gegeben Teilung gewinnbringend ist, anhand von Gl. 4.4 erfolgen. Diese Entscheidung könnte prinzipiell auch über mehrere Hierarchieebenen, also mehr als den unmittelbaren Vater-zu-Kindknoten Bezug hinweg getroffen werden [Laz96, ChC97]. Durch die Verwendung klassenspezifischer Crossvalidierungsdaten lässt sich die Baumstruktur an die jeweilige Sprechgeschwindigkeitsklasse anpassen. Mit einer nachfolgenden Trainingsiteration (vgl. Abb. 4.3) mit den gesamten bzw. den klassenweisen Trainingsdaten kann eine Anpassung der Mixturgewichte erfolgen. Für die eigentliche Erkennungsphase sind dann nur noch die neu entstandenen Blattknoten von Interesse.

Auffallend an den Wortfehlerraten in Tab. 4.15 ist die deutliche Verschiebung der WER zwischen den unterschiedlichen SR-Kategorien der Testdaten - bei annähernd gleicher Gesamtfehlerrate. Dies bestätigt wiederum, dass bei ausschließlich zustandsabhängigen Codebüchern die Struktur des Entscheidungsbaums starken Einfluss auf die sprechgeschwindigkeitsspezifische Erkennungsleistung hat.

SR Kat. CV-/Train.daten		WER, SR Kategorie der Testdaten					
Pruning	Nachtraining	Gesamt	sehr l.	langsam	mittel	schnell	sehr s.
langsam	gesamt	24.9	20.0	22.0	23.3	27.1	35.0
mittel	gesamt	24.7	21.5	24.3	22.6	26.0	34.9
schnell	gesamt	24.8	20.6	26.9	22.4	26.7	32.6
langsam	langsam	25.4	21.1	24.2	24.3	26.0	36.0
mittel	mittel	24.6	21.1	24.2	22.6	26.4	34.9
schnell	schnell	25.1	21.0	24.7	23.2	27.3	34.9
Basissystem		24.7	22.7	24,9	22,6	27.1	31,4

Tab. 4.15: WER (in [%]) der aus den Knotenbäumen entstandenen Blattbäume. Das Pruning erfolgte mit kategorieweisen CV-Daten, das Nachtraining mit den gesamten bzw. den kategorisierten Trainingsdaten.

4.4 Entscheidungsbäume mit generalisiertem Kontext

4.4.1 Generalisierter Kontext

In diesem Abschnitt soll eine Erweiterung des phonetischen Entscheidungsbaumverfahrens vorgestellt werden, um auch nicht-phonetische Fragen im Entscheidungsprozess verwenden zu können. Der Schwerpunkt wird hierbei auf die Integration sprechgeschwindigkeitsspezifischer Information gelegt. Vorgeschlagen wurden derartige Erweiterungen des Entscheidungsbaumkonzepts bereits von Paul [Pau97] oder Reichl [ReW99]. Paul versuchte in seiner Arbeit die strikte Zustandsbindung der Entscheidungsbäume aufzulösen, indem er Fragen nach den Basisphonemen, sowie den jeweiligen HMM-Zuständen (sog. 'Single-Tree') zuließ. Der Übergang zu einem einzelnen, globalen Entscheidungsbaum erbringt allerdings keine nennenswerte Verbesserung der Erkennungsleistung [Laz96, Pau97]. Reichl hingegen untersuchte die Berücksichtigung eines geschlechtsspezifischen Kontexts im Entscheidungsprozess - allerdings nur für den Fall eines bekannten Kontexts ('supervised'). Parallel zu dieser Arbeit wurde auch von Fügen [Fue00] ein ähnlicher Ansatz zur Integration von Sprechgeschwindigkeitsinformation vorgestellt. Eine vergleichende Diskussion dieser Arbeit folgt im Anschluss. Betrachtet werden im folgendem insbesondere die Auswirkungen auf die Struktur des Erkennungssystems, bzw. notwendige Modifikationen und Veränderungen im Vergleich zu einem rein 'statischen' Suchvorgang.

Unter dem Begriff 'Generalisierter Kontext' ist diejenige Art von Kontext zu verstehen, die nicht ausschließlich phonetischer Natur ist. Im allgemeinen wird bei Entscheidungsbaumverfahren zur Zustandsgruppierung primär phonetischer Kontext verwendet, d.h. die Information in welcher Laut-Nachbarschaft ein Phonem auftritt. Bei Triphonen wird hierbei nur das unmittelbare rechte bzw. linke Nachbarphonem in die Kontextbetrachtung einbezogen. Generalisierter Kontext hingegen beinhaltet auch allgemeine Fragen nach dem nicht-phonetischen Umfeld, u.a.:

- Geschlecht (männlich/weiblich)

- Sprecher bzw. Sprechergruppe
- Sprechgeschwindigkeit
- Umgebungsgeräusch
- Dialekt, geograph. Region des Sprechers
- Position des Lauts im Wort, im Satz etc.

Die weiteren Untersuchungen konzentrieren sich schwerpunktmäßig auf die Kontexte, die als sogenannte "dynamische" Kontexte aufgefasst werden können. Diese Definition bezieht sich auf ursächliche Eigenschaften des akustischen Signals, die sich im Laufe einer Spracheingabe ändern können. Sprechgeschwindigkeit ist hierfür das hervorstechendste Beispiel. Informationen über den Sprecher (z.B. Sprecher(gruppe), Dialekt, Region) stellen bzgl. eines einzelnen Turns einen statischen Kontext dar, da keine Änderung desselben eintritt. Bei etwas allgemeinerer Betrachtungsweise kann diese Information jedoch als ebenso dynamisch aufgefasst werden, da die Beschränkung auf einen sprechersegmentierten Einzelturn nicht zwingend vorliegen muss und i.a. auch nicht vorliegt. So können durchaus mehrere Personen abwechselnd an einer Spracheingabe beteiligt sein. Beispiele hierfür sind Dialoge, oder die sprachliche Steuerung von Bediengeräten im Kfz. Sie können sowohl von Fahrer als auch Beifahrer adressiert werden.

Die Auffassung der genannten Umgebungsbedingungen als "Kontext" ist an dieser Stelle motiviert durch den Einsatz von Entscheidungsbäumen. Allgemein gesehen sind diese laufzeitabhängigen Bedingungen Einflussgrößen auf die akustischen Merkmalsvektoren und damit direkt auf die Übereinstimmung mit den statistisch geschätzten Modellen. Gesucht ist in diesem Zusammenhang eine selektive Auswahl des Modells bzw. der Modellstrukturen, mit dem Ziel die Abweichung zu reduzieren.

Vorteile: Viele der oben angeführten allgemeinen Fragen, z.B. Geschlecht, bedingen eine harte Einteilung in Klassen. Andere, wie z.B. Sprechgeschwindigkeit, sind zwar von Natur aus kontinuierlich (wenngleich auch die Messung ein nicht-triviales Problem darstellt), lassen sich durch eine Quantisierung jedoch ebenfalls in Kategorien fassen (vgl. Abschnitt 3.3.1). Eine separate Modellbildung würde, speziell bei der Kombination mehrerer dynamischer Kontexteigenschaften, zu einem starken Absinken des individuellen verfügbaren Trainingsmaterials führen, da für eine einzelne Modellgruppe nur noch ein Bruchteil des ursprünglichen Trainingsmaterials verwendet werden kann. Eine robuste Schätzung der Modellparameter wird dadurch, speziell für Phonemmodelle mit bereits begrenztem Datenmaterial, zunehmend kritisch. Erschwerend kommt hinzu, dass einige der Kontexteigenschaften, aufgrund der nichtvorliegenden Gleichverteilung (z.B. Sprechgeschwindigkeit), eine ungleichmäßige Datenaufteilung bedingen. Bei noch weitreichenderen (dynamischen) Kontextbetrachtungen scheidet eine separate Modellbildung bei einem gleichzeitigen ML-Training aller Parameter damit de facto aus. Abhilfe kann hier durch die Reduktion der freien, zu schätzenden Parameter geschaffen werden. Dies kann beispielsweise durch den Verzicht auf die Anpassung aller Parameter geschehen: typisch hierfür wäre es, nur die Mittelpunktsparameter nachzuführen. Ein anderer

Weg ist die Anwendung von gemeinsamen Parametertransformationen (z.B. MLLR). Die Gesamtzahl der freien Parameter der Transformationsmatrizen ist relativ gering, wodurch sich diese vergleichsweise robust schätzen lassen.

Darüber hinaus ist für manche Phoneme eine Einteilung in separate Kategorien gar nicht notwendig - u.U. sogar schädlich - da sie durch den dynamischen Kontext nur in geringerem Maße beeinflusst werden. So sind beispielsweise Plosive im Vergleich zu Vokalen, speziell im Hinblick auf die Phonemdauer, von der Sprechgeschwindigkeit weit weniger beeinflusst [Kuw96, Kuw97, Mar97]. Durch die Einbeziehung in den Entscheidungsbaum (DynKon-Bäume) wird diese vollständige Trennung der Modellgruppen untereinander aufgehoben. Eine Trennung erfolgt - gemäß Gütemaß - nur dort, wo auch ein 'Modellierungsvorteil' erkennbar ist.

Nachteile: Durch die Einbindung dynamischen Kontexts in das Entscheidungsbaumverfahren, werden die Baumstrukturen - bei gleichen Abbruchbedingungen, d.h. N_P^{min} und L_C - meist größer (s. Tab. 4.16) als die vergleichbaren Bäume ohne dynamischen Kontext. Dies ist insofern nicht als ausschließlicher Nachteil zu sehen, da sich hierdurch bei gleicher Gesamtzahl der Verteilungsparameter, d.h. einer reduzierten Anzahl von Verteilungen je Codebuch, Vorteile in der Verarbeitungsgeschwindigkeit (=Rechenzeit) ergeben. Darüber hinaus zeigt dies, dass die Einteilung im Mittel für einige Phoneme von Vorteil ist.

gen. Kontext.	N_B
kein	3506
Geschlecht	4830
SR	4220
Geschlecht+SR	5223

Tab. 4.16: Gesamtzahl der Blattknoten aller Bäume bei $L_C = 300$ und $N_P^{min} = 250$.

Nicht zu vernachlässigende Nachteile ergeben sich jedoch durch die notwendigen Modifikationen des Erkennungssystems, sowie durch den erhöhten Aufwand in der Erkennungsphase. Ähnliche Anmerkungen äußerte auch Fügen in [Fue00], der durch die Randbedingungen des von ihm verwendeten JANUS-Erkennungssystems die Baumstruktur modifizieren musste, um die zusätzliche Information integrieren zu können. In dieser Arbeit konnte eine direkte Integration realisiert werden, da bei der Implementierung - im Gegensatz zum JANUS-System - keine systemtechnischen Einschränkungen vorlagen. Kernpunkt der Einbettung in das Erkennungssystem ist die Zuordnung eines beliebigen Lexikonknotens zum zu verwendenden Modell. Das Lexikon ist in baumförmiger Struktur [Pla95] aufgebaut und verweist für jeden Triphoneintrag des Lexikons eindeutig auf das zugehörige Modell bzw. die zugehörige Verteilung (s. Abb. 4.7).

Um für ein beliebiges Triphon mit gegebenem generellen Kontext für einen bestimmten Zustand das korrekte Codebuch zu ermitteln, müsste im Prinzip der Entscheidungsbaum durchlaufen (=Parsing) werden. Bei Entscheidungsbäumen ohne generellen Kontext kann

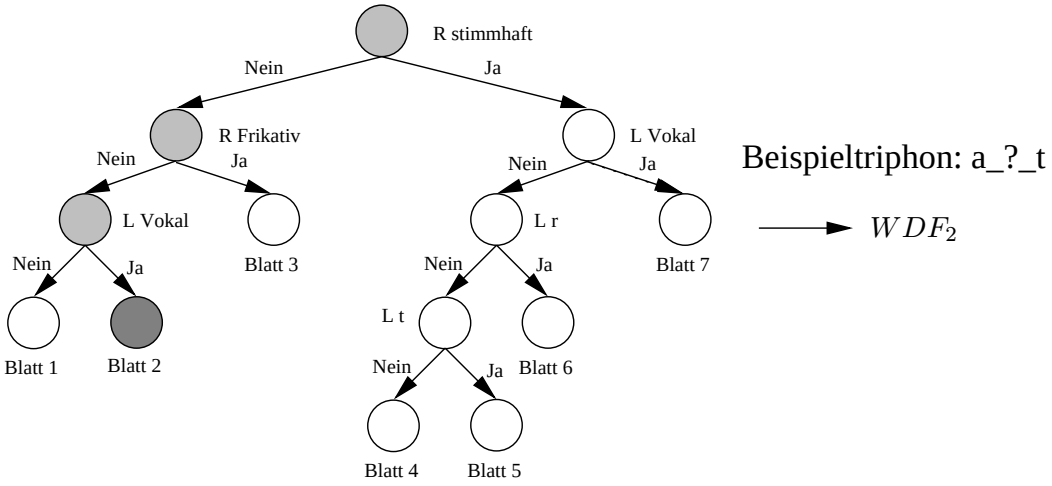

Abb. 4.7: Zuordnung von Lexikoneintrag zu Modell bzw. Codebuch bei rein statischen Entscheidungen.

dies offline, d.h. im voraus geschehen, da für jeden Lexikoneintrag nur eine einzige Zuordnung (je Zustand) nötig ist. Allgemein verbietet sich ein Parsing des Entscheidungsbaums zur Laufzeit jedoch mit Blick auf den Rechenaufwand. Versuche zeigten, dass hierdurch der Realzeitfaktor, verglichen mit der offline Bestimmung der Zuordnung, um ca. 85% ansteigt. Bei der Einführung genereller Kontexte ist die Zuordnung eines Lexikoneintrags zum Modell nicht mehr eindeutig, sondern hängt vom aktuell vorliegenden dynamischen Kontext ab. Abb. 4.8 zeigt dies anhand eines Beispielbaums.

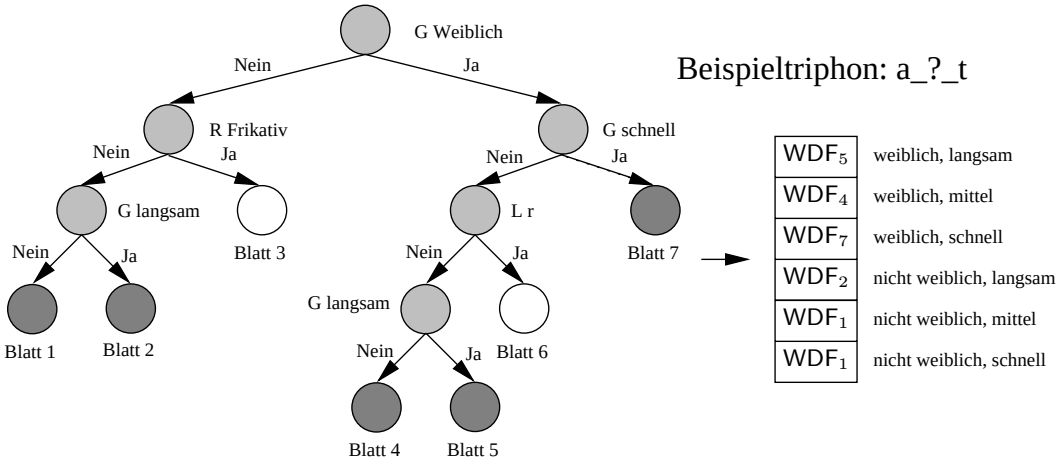

Abb. 4.8: Dynamische Entscheidungen: Zuordnung von Lexikoneintrag zuWDF abhängig von dynamischem Kontext.

Beim Durchlaufen ist der letztendlich zugeordnete Terminalknoten abhängig vom vorliegenden generellen Kontext. In obigem Beispiel ergeben sich bei der Verwendung von 6 Kontextkombinationen (3 für Sprechgeschwindigkeit, 2 für Geschlecht) 6 potentielle Endknoten. In der Praxis treten meist wesentlich weniger in Frage kommende Blattknoten auf.

Nichtsdestoweniger muss zur Laufzeit immer noch eine Auswahl aus diesen $N_L \leq 6$ Knoten eine Auswahl getroffen werden. Dies stellt insofern eine Rechenzeitbelastung dar, als bei der Suche in jedem Frame für alle aktiven Hypothesen diese Zuordnung vorliegen muss. Die Zahl der Abfragen übersteigt damit sehr leicht 100000/Frame. Wenn bei jeder Abfrage erst eine Suche eingeleitet werden muss, hat dies äußerst negative Auswirkungen auf die Verarbeitungsgeschwindigkeit.

Fügen [Fue00] schlägt in diesem Zusammenhang eine Restrukturierung des Entscheidungsbaums vor. Durch eine geschickte Vertauschung der Fragen wird es ihm möglich, alle bauminternen Knoten, die einen generellen Kontext betreffen, nach "außen" zu schieben und daher in Blattknoten zu konvertieren. Nach der Umschichtung ist ein Teil der Blattknoten sensitiv bzgl. des generellen Kontexts, der Rest bleibt unabhängig. Auf diese Weise ist nur für einen Teil der Blattknoten die Suche in einer Lookup-Tabelle wirklich notwendig. Die ausschlaggebenden Gründe für die Umstrukturierung sind allerdings Systemeinschränkungen des von Fügen verwendeten JANUS-Systems. Um den Verwaltungsmehraufwand zu kompensieren, wurde der vorliegenden Arbeit ein zweistufiges Vorgehen implementiert. Um die Suche innerhalb des Baums gänzlich zu vermeiden wurde ein Zugriff über Lookup-Tabellen realisiert. Im Gegensatz zu Fügen, der den Zugriff auf Blattebene angesiedelt hat, wurden die Lookup-Tabellen in dieser Arbeit triphonweise organisiert. Hierzu werden beim Start des Systems für jeden Zustand eines Triphon-Lexikoneintrags die möglichen Blattreferenzen ermittelt. Da einerseits nur die im Lexikon vorkommenden Triphone betrachtet werden müssen und anderseits nicht alle dieser Triphone generell-kontextsensitiv sind, fällt der zusätzliche Speicherbedarf sehr gering aus. Um den Rechenzeitmehraufwand, der durch die Adressindexierung der Lookup-Tabellen verursacht wird, zu umgehen, wurde eine Cache-Strategie entwickelt und implementiert. Die zuletzt verwendete, generalisiert-kontextsensitive Zuordnung eines Triphons wird in einem triphonspezifischen Cachespeicher zwischengespeichert. Eine Neubestimmung der Blattknoten ist dann nur noch notwendig, wenn eine Änderung des generellen Kontexts eintritt. Da eine Änderung desselben nur vergleichsweise (mit der Zahl der Frames) selten auftritt, kann hierdurch der, durch die Einführung des generellen Kontexts, entstandene Mehraufwand stark eingeschränkt werden.

Die Zuordnung erst zur Laufzeit bzw. zum Laufzeitbeginn birgt noch einen weiteren systemtechnischen Vorteil: Das Lexikon wird unabhängig vom verwendeten nicht-phonetischen Kontext. Prinzipiell könnte die Zuordnung auch in die Lexikonstruktur integriert werden, was mit Blick auf den Rechenzeitbedarf geringfügige Vorzüge böte. Im Gegensatz dazu, kann durch die Zuordnung erst zur Laufzeit das Lexikon dynamisch gehalten werden. Speziell für Dialogsysteme kann es notwendig sein, das Lexikoninventar - bedingt durch gewisse Dialogsituationen - zu erweitern oder verringern.

4.4.2 Ergebnisse

4.4.2.1 Geschlecht

Die separate Modellierung des Geschlecht ist eine intuitive Einteilung. Dass jedoch eine vollständig getrennte Modellierung nicht sinnvoll sein muss, zeigt folgende Tabelle:

	separat	DynKon	ohne
männl.:	25,7	26,8	26,2
weibl.	29,2	23,0	25,4

Tab. 4.17: WER für getrennte Geschlechts-Modelle (ML-trainiert) und DynKon-Bäume.

Erstaunlicherweise sind die Ergebnisse etwas zwiespältig: für das männliche Geschlecht haben die DynKon-Bäume eine ca. 1% absolut schlechtere Performanz, aber im Gegensatz dazu eine 6% absolut bessere Performanz für das weibliche Geschlecht. Diese Ergebnisse gelten allerdings unter idealen Annahmen, d.h. sie setzen eine 100% - korrekte Zuordnung von Modellgruppe (m/w) und aktuellem Sprecher(m/w) voraus. Allerdings ist die m/w-Zuordnung vergleichsweise unkritisch. Mit GMM-basierten Klassifikationssystemen kann eine hohe Sicherheit bei der m/w-Entscheidung erreicht werden (vgl. Tab. 3.7). Fehlklassifikationen treten meist nur bei sehr kurzen Sprachsequenzen (Bsp. “Guten Tag”) auf [He99a]. Gerade bei so kurzen Sequenzen hat die Falscheinstufung bzgl. der Spracherkennung meist nur geringe Auswirkungen.

4.4.2.2 Sprechgeschwindigkeit

Ein besonderer Aspekt der Integration von Sprechgeschwindigkeitsklassen in den Entscheidungsprozess ist die Auswahl bzw. Einteilung der Trainingsdaten. Wie Abb. 4.4 beispielhaft zeigt, sind die Sprechgeschwindigkeiten annähernd gaussverteilt. Dies gilt sowohl bei Auswertung der Sprechgeschwindigkeit auf Satzebene, Spurtebene oder bei Auswertung der lokalen Sprechgeschwindigkeit. Von Bedeutung ist daher die Festlegung der Klassengrenzen für die Einteilung der Trainingsdaten.

$$C_{ROS_{Ph}} = \begin{cases} \textit{schnell} & \text{wenn } v_{Turn} > \mu_{ROS_{Ph}} + \Delta \\ \textit{langsam} & \text{wenn } v_{Turn} < \mu_{ROS_{Ph}} - \Delta \\ \textit{mittel} & \text{sonst} \end{cases}$$

Im Gegensatz zum Vorgehen von Fügen in [Fue00] wurden dem realisierten Algorithmus 3 Sprechgeschwindigkeitsklassen vorgegeben. Fügen verwendete hier nur eine 2-Klassen Einteilung mit $v_{SR} < v_{Schwelle}$ bzw. $v_{SR} \geq v_{Schwelle}$, wobei die Sprechgeschwindigkeit von ihm anhand des mrate-Maßes [Mor98] bestimmt wurde. Es ist jedoch vorteilhafter, alle 3 Klassen zur Entscheidung zuzulassen, da es damit dem Algorithmus überlassen wird, auszuwählen, welche Klasse bzw. Einteilung von Bedeutung ist.

Bei der angegebenen 3-Klassen Einteilung der Daten und $\Delta = \sigma_{ROS_{Ph}}$ entfallen nur jeweils ca. 15% der Daten auf die Randkategorien schnell und langsam. Aufgrund der unteren Grenze für die Zahl der Trainingsvektoren in einem Blatt wird ein Großteil der Fragen nach den Sprechgeschwindigkeitsrandkategorien ausgefiltert.

Tabelle 4.18 zeigt, dass in den entstehenden Entscheidungsbäumen nur ein geringer Bruchteil aller ausgewählten Fragen eine Sprechgeschwindigkeitsrandkategorie betreffen. Bei genauerer Aufschlüsselung richtet sich von diesen wenigen Fragen die Mehrzahl nach langsamer

	berücksichtigte Kontextklassen	
Frage nach	nur SR	SR+Geschlecht
SR langsam	59	37
SR mittel	509	480
SR schnell	6	3
SR total	574	520
beliebig (alle)	3995	5200

Tab. 4.18: Anzahl der Fragen nach Sprechgeschwindigkeit für $\Delta = \sigma_{ROS_{Ph}}$.

Sprechgeschwindigkeit. Hintergrund hierbei ist, dass langsame Sprachsegmente meist mehr Sprachvektoren aufweisen, als entsprechende schnelle Sprachabschnitte. Erstaunlicherweise ist die Frage nach mittlerer Sprechrate sehr häufig anzutreffen. Bei der Auswertung der Likelihood dieser Frage wird in 'mittel' und 'nicht-mittel' geteilt, d.h. die Daten der Kategorien 'schnell' und 'langsam' werden gemeinsam bewertet. Durch diese Kombination ergeben sich bei einer Teilung offensichtlich genügend Restvektoren um die Teilungen gemäß des N_P^{min}-Kriteriums zu erlauben.

Um die Auswirkungen, bzw. das Auftreten von Fragen bzgl. des Sprechgeschwindigkeitskontexts näher zu untersuchen kann die Klassenschwelle abgesenkt werden. Bei einer Einstellung von $\Delta = (1-\epsilon)\sigma_{ROS_{Ph}}$ mit $\epsilon = \frac{1}{2}$ zeigt sich eine veränderte Auftretenshäufigkeit diesbezüglicher Fragen (s. Tab. 4.19).

	berücksichtigte Kontextklassen		
Frage nach	nur Geschlecht	nur SR	SR+Geschlecht
SR langsam	-	462	452
SR mittel	-	101	65
SR schnell	-	103	94
SR total	-	666	611
Geschlecht	446	-	447
Wortgrenze	312	257	304

Tab. 4.19: Gesamte Zahl der Fragen nach Sprechgeschwindigkeit bei $\Delta = (1-\epsilon)\sigma_{ROS_{Ph}}$ bei Bäumen mit generalisierten Entscheidungen.

Verglichen mit Tabelle 4.18 treten mehr Fragen nach der Sprechrate auf. Bei den Teilungen verbleiben also häufiger ausreichend Vektoren ($N_P^{ja}, N_P^{nein} > N_P^{min}$) in den Kindknoten, was die Durchführung der Teilungen aus dieser Hinsicht zulässt. Andererseits findet eine ausgeprägte Verschiebung der Fragen hin zu langsamer Sprache statt. Interessant ist in diesem Zusammenhang der Stellenwert dieser Entscheidungen, v.a. im Vergleich zur Frage nach dem Geschlecht. Kennzeichnend für die Bedeutung ist insbesondere das erste Auftreten einer solchen Frage im Entscheidungsbaum. Tabelle 4.20 schlüsselt dieses Kennzeichen für repräsen-

tative Phoneme auf. Angegeben sind u.a. die Knotenschicht, in der die jeweilige Frage zum ersten Mal auftritt, sowie die Gesamtzahl der Schichten N_L und die Gesamtzahl der Knoten N_K. Die beiden letzteren Informationen geben Aufschluss darüber, wie gleichmäßig der Baum strukturiert ist. Bei einem optimal strukturierten Binärbaum sollte gelten: $N_K = 2^{N_L} - 1$ bzw. $N_L = \log_2(N_K + 1)$ (s. Abb. 4.9), d.h jeder Knoten mit Nachfolgern hat sowohl im Ja- als auch im Nein-Pfad die gleiche Anzahl Nachfolgerknoten.

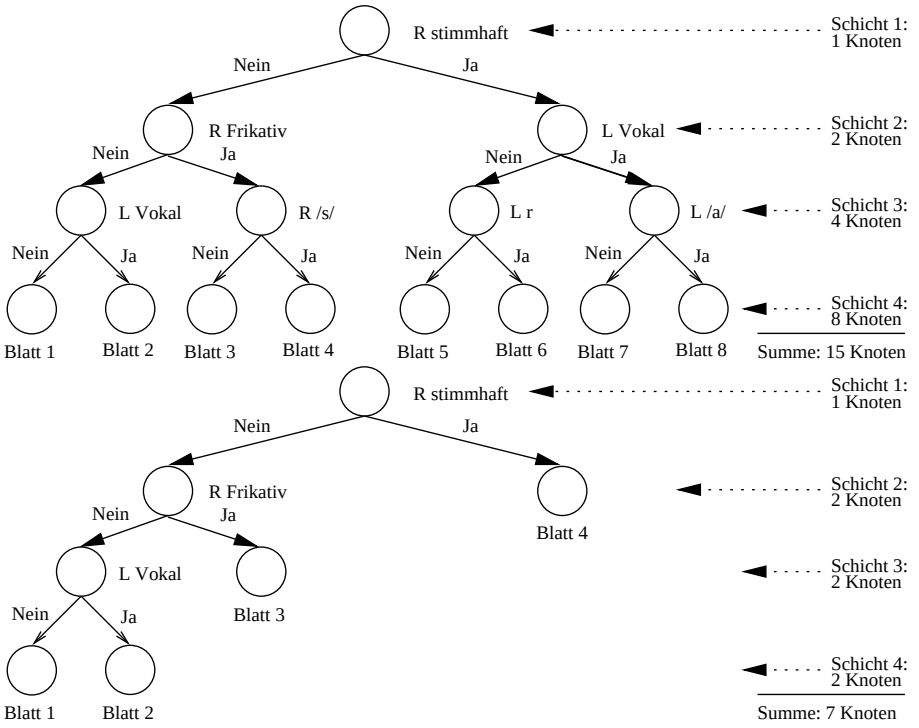

Abb. 4.9: Schichten bei einem optimal symmetrischen Binärbaum (oben) und einem stark asymmetrischen Baum (unten). Beide Bäume haben 4 Schichten, jedoch hat der obere Baum insgesamt 15 Knoten, der untere nur 7.

Tabelle 4.20 macht deutlich, dass die Geschlechtsinformation weitreichendere Auswirkungen hat. Bei vielen Phonemen tritt die Frage nach dem Geschlecht sehr nahe am Wurzelknoten (1.Schicht, s. Abb. 4.9) auf, d.h. als allererste Einteilung. Bemerkenswert ist hierbei, dass dies vorwiegend für Vokale und Frikative der Fall ist. Von untergeordneterer Bedeutung scheint das Geschlecht für Plosive und Nasale zu sein. Im Vergleich zur Geschlechtsinformation treten Fragen nach der Sprechgeschwindigkeit tendenziell eher später - entfernter vom Wurzelknoten - auf. Im Mittel finden sich diese Fragen erstmalig etwa in der 3. bis 4. Schicht. Diese Feststellung muss natürlich immer im Zusammenhang mit der Festlegung der Klassengrenzen der Sprechgeschwindigkeitskategorien gesehen werden.

4.4.2.3 Erkennungsergebnisse

Anders als bei den biologisch fundierten Geschlechterkategorien (wenngleich auch manche Sprecher weiblich klingende Stimmen haben - und umgekehrt), sind die Klassengrenzen bei

	Geschlecht			SR		
Phonem	s_1	s_2	s_3	s_1	s_2	s_3
/z/	1	1	1	-	3	4
/s/	2	1	2	6	4	6
/a/	3	2	2	6	3	6
/E:/	2	2	2	4	3	4
/m/	3	3	3	3	3	4
/f/	2	1	4	2	4	6
/t/	3	4	3	5	6	6
/p/	3	3	4	-	4	-

Tab. 4.20: Schicht des ersten Auftretens von Fragen nach dem dynamischen Kontext für die gruppierten Triphonzustände $s_1..s_3$.

Sprechgeschwindigkeit nur per Definition vorgegeben. Dies gilt, wie bereits erläutert, sowohl für die Einteilung während des Trainings bzw. der Baumgenerierung, als auch bei der Erkennung. Für die Erkennung ist daher ebenfalls eine Bestimmung der Sprechrate notwendig.

Im Rahmen dieser Arbeit wurden verschiedene Ansätze zur Bestimmung der Sprechgeschwindigkeit während bzw. nach der Erkennung untersucht. Die Referenzrate für jeden Spurt der Äußerung lässt sich anhand einer phonetischen Segmentierung ermitteln. Verwendet wurde in den Experimenten die Sprechgeschwindigkeitsdefinition aus Kapitel 3.2. Wenn der Inhalt der Äußerung bekannt ist (Transliteration), dann entspricht dies einem 'cheating experiment'. Anders verhält es sich, wenn - wie im realen Anwendungsfall gegeben - deren Inhalt nicht bekannt ist. In diesem Fall kann die Sprechgeschwindigkeit anhand einer Erkennerhypothese ermittelt werden [Sie95, Mir96], die in einem ersten Durchgang generiert wird. Da im Prinzip nur die Zahl der Phoneme innerhalb gewisser Grenzen von Interesse ist, kann dieser erste Durchlauf auch eine rein freilaufende Phonemerkennung sein. Ausgehend von der auf diese Weise gewonnenen Information über die Sprechgeschwindigkeit kann, in einem zweiten Durchgang, die Erkennung mit den entsprechenden sprechgeschwindigkeitsspezifischen Modellen durchgeführt werden. Ein ähnliches 2-stufiges Vorgehen findet sich auch bei Fabian in [Fab01]. Er verwendet die anhand der Phonemsequenz ermittelte Sprechgeschwindigkeit, um die N-Besten Liste des Erkenners passend umzusortieren.

Die alleinige Berücksichtigung des Sprechgeschwindigkeitskontexts erlaubt nur eine geringfügige Verbesserung der Worterkennungsrate. Im Zusammenspiel mit dem Geschlechtskontext kann die WER im Vergleich zum Ausgangssystem deutlich reduziert werden (s. Tab. 4.21 und 4.22). Die Reduktion ist hierbei stärker als bei alleiniger Verwendung einer Kontextinformation. Nachteil des 2-stufigen Ansatzes zur Sprechgeschwindigkeitsbestimmung ist der nötige zweite Durchgang. Bei einem System, dessen Realzeitfaktor größer als 1 ist, bedeutet dies einen erhöhten zeitlichen Aufwand, d.h. mehr Wartezeit für den Anwender. Um diesen zu vermeiden, wurde ein GMM-basierter Ansatz entwickelt, der es erlaubt, die Sprechgeschwindigkeit während des ersten Durchlaufs online zu schätzen. Eine Beschreibung bzw. Diskussion dieses Ansatzes kann Kapitel 3.3.1 entnommen werden. Die Sprechgeschwindig-

		WER, SR Kategorie der Testdaten					
berücks. Kontext	gesamt	sehr l.	langsam	mittel	schnell	sehr s.	N_B
kein	25.8	22.3	28.1	23,5	27.1	31.5	3506
Geschlecht	24.7	22.7	24.9	22.6	27.1	31.4	4830
SR	27.1	21.9	27.4	25.4	30.7	32.6	4220
Geschlecht+SR	24.5	22.1	24.7	22.3	27.2	31.1	5432

Tab. 4.21: WER für Aussprachevarianten-basierte Modelle mit Berücksichtigung unterschiedlicher genereller Kontexte im Entscheidungsbaum.

		WER, SR Kategorie der Testdaten					
berücks. Kontext	gesamt	sehr l.	langsam	mittel	schnell	sehr s.	N_B
kein	26.6	22.3	28.6	25.4	27.2	30.1	3022
Geschlecht	25.2	21.5	26.9	23.9	25.6	29.6	4326
SR	26.2	21.0	28.1	24.7	27.5	30.7	3807
Geschlecht+SR	24.4	23.2	25.3	22.7	24.8	30.1	5056

Tab. 4.22: WER für kanonische Modelle mit Berücksichtigung unterschiedlicher genereller Kontexte im Entscheidungsbaum.

keit kann damit parallel zum eigentlichen Suchvorgang ermittelt werden und steht diesem in jedem Frame zur Verfügung. Der Mehraufwand durch die 3 zu berechnenden GMM-Modelle ist vergleichsweise gering.

Die beiden obigen Ansätze implizieren einen theoretischen Nachteil: durch eine frühe Falscheinschätzung der Sprechgeschwindigkeitskategorie wird u.U. vorzeitig die optimale Kategorie verworfen. Vorteilhafter wäre in dieser Hinsicht die Auswertung aller möglichen Suchpfade mit einer Entscheidung - einschließlich der Entscheidung über die Sprechgeschwindigkeit - erst am Ende der Spracheingabe (vollständig implizit). Ohne weitergehende explizite Einschränkungen für die Übergänge zwischen Kategorien [Zhe00], kann in jedem Frame prinzipiell jede Kategorie vorliegen. Es müssen also alle möglichen Kategorien C_v bewertet werden:

$$p(\boldsymbol{x}_j|m,s) = \max_{C_v} p(\boldsymbol{x}_j|m,s,C_v) \tag{4.10}$$

Ohne Übergangseinschränkungen kann die implizite Suche auf eine lokale Maximum-Entscheidung reduziert werden. Nachteil des impliziten Ansatzes ist jedoch der erhöhte Aufwand zur Mehrfachberechnung der Emissionen aller Kategorien. Bei der Maximierung in Gl. 4.10 müssen daher $|C_v|$ Codebücher berechnet werden.

Bemerkenswert in Tabelle 4.23 ist die relative Unabhängigkeit der ermittelten Erkennungsraten von der Qualität der Sprechgeschwindigkeitsbestimmung. Mit einer Online-Schätzung der Sprechgeschwindigkeit kann eine ähnliche, teilweise sogar bessere Performanz erzielt werden, als durch die Sprechgeschwindigkeitsbestimmung anhand der Referenztranskription. Einzig bei einer vollständig impliziten Bestimmung ergibt sich deutlicher Abfall der Erken-

			WER, SR Kategorie der Testdaten				
berücks. Kontext	SR Messung	gesamt	sehr l.	langsam	mittel	schnell	sehr s.
SR	Translit.	26.2	21.0	28.1	24.7	27.5	30.7
SR	GMM	26.1	22.9	27.4	24.9	27.5	28.8
SR	vollst. implizit	28.3	24.8	28.4	27.6	29.3	33.0
Geschlecht+SR	Translit.	24.4	23.2	25.3	22.7	24.8	30.1
Geschlecht+SR	GMM	24.0	22.3	25.6	23.6	23.6	28.8

Tab. 4.23: WER für kanonische Modelle bei unterschiedlicher Messung der vorliegenden Sprechgeschwindigkeit (GMM: 3 GMMs mit je 16Bf, MFCC42).

nungsrate. Zurückzuführen ist dies auf die gestiegene Verwechselbarkeit zwischen den Phonemen, da die Auswahl anhand des besten Scores in jedem Frame erfolgt.

Vergleicht man die verschiedenen untersuchten Ansätze bezüglich ihrer Leistungsfähigkeit, so lässt sich zusammenfassend feststellen, dass insbesondere die Integration des Sprechgeschwindigkeitskontexts in den Entscheidungsprozess zu einer deutlichen Reduktion der Fehlerrate führt. Diese wird jedoch erst durch die Kombination mit dem Geschlechtskontext voll wirksam. Ähnlich wirkungsvoll zeigt sich die Optimierung der Baumstruktur durch eine geeignete Pruningstrategie. Eine sehr einfache, schnelle und wirkungsvolle Maßnahme zur Kompensation des Sprechgeschwindigkeitseinflusses zeigt sich in der Anpassung des Sprachmodellfaktors.

Kapitel 5

Automatische Sprechergruppierung

5.1 Einführung

Die Variation in den sprecherspezifischen Sprachcharakteristika stellt für die automatische Spracherkennung noch immer eines der größten Probleme dar. Wünschenswert wäre ein sprecherunabhängiges (SI: engl.: 'speaker independent') Erkennungssystem, das für jeden Anwender gleichermaßen gut funktioniert. In der Praxis hat sich jedoch gezeigt, dass SI-Systeme zwar für eine größere Sprecherpopulation im Mittel gut funktionieren, für einen speziellen Sprecher jedoch meist schlechter als ein entsprechendes sprecherspezifisches (=sprecherabhängig, SD, engl.: 'speaker dependent'). Bei einem SD-System werden die Systemparameter durch ausschließliche Verwendung von Trainingsäußerungen des Zielanwenders speziell für diesen trainiert. Im Unterschied hierzu werden beim Training eines SI-Erkennungssystems die Modellparameter aus einem Sprachkorpus von möglichst *vielen* Sprechern geschätzt. Ein SD-Erkenner arbeitet daher meist für alle anderen Sprecher deutlich schlechter als ein entsprechendes SI-System.

Eine der Primärzielsetzungen der Forschung war - und ist - ein ideales SI-System zu realisieren, das für alle Nutzer gleich optimal funktioniert. Dieses Ziel scheint allerdings durch das alleinige Training mittels Multi-Sprecherkorpora nicht realisierbar zu sein. Die Forschungsanstrengungen konzentrieren sich nunmehr verstärkt darauf, wie sich "sub-optimale" SI-Systeme gezielt an den jeweiligen Anwender anpassen lassen. Die Idee hierbei ist eine Kombination aus sprecherunabhängigem, robustem Parametertraining mit anschließender Anpassung an den/die Zielsprecher.

Eine vergleichende Untersuchung zwischen sprecherabhängiger und -unabhängiger Modellerzeugung, wurde von Huang in [Hua91] veröffentlicht. In den beschriebenen Experimenten wiesen die sprecherabhängigen Modelle eine um bis zu 40% relativ geringere Wortfehlerrate auf. In dieser Publikation wird jedoch ein Problem bei der Erstellung von SD-Modellen deutlich: für die Erzeugung der akustischen Modelle ist eine hohe Anzahl von Trainingsäußerungen des Zielsprechers nötig. Huang spricht hier von mehr als 600. Eine solch beträchtliche Zahl an Trainingsdaten steht jedoch bei den meisten Anwendungen entweder nicht zur Verfügung, oder es kann dem Anwender nicht zugemutet werden sie einzugeben. Gerade die Frage der Benutzerakzeptanz ist bei Spracherkennungssystemen - insbesondere bei Diktiersy-

stemen - von wirtschaftlicher Bedeutung. Theoretisch ist es zwar möglich, dass ein Benutzer umfangreiche Trainingsdaten eingibt - in der Praxis sinkt damit jedoch die Bereitschaft der Anwender solche Systeme einzusetzen deutlich.

Bei anderen Anwendungen, wie beispielsweise automatisierten (Telefon)Auskunftsystemen, ist die Erstellung von SD-Modellen schlicht nicht möglich bzw. sinnvoll. Die Wahrscheinlichkeit, dass ein bestimmter Sprecher das System wiederholt benutzt, ist extrem gering. Darüber hinaus ist die mittlere Anrufdauer gerade bei Auskunftssystemen oft kleiner als 1 Minute - für ein explizites ML-Training deutlich zu wenig. Hinzu kommt im allgemeinen eine weitere Schwierigkeit: die Sprachdaten müssen für ein überwachtes Training verschriftet und segmentiert werden. Manuelle Verschriftung kommt i.d.R aus Zeit- und Personalgründen nicht in Frage und ist meist nur dann möglich, wenn der Zielanwender zum Zeitpunkt der Systemerstellung bekannt ist. Handelt es sich jedoch um ein offenes System, dessen Anwender zu diesem Zeitpunkt also nicht bekannt sind, so müssen andere Möglichkeiten zur Verschriftung angewandt werden.

Bei Diktiersystemen geschieht dies meist dadurch, dass dem Benutzer der zu sprechende Text vorgegeben wird. Bei anderen Applikationen ist ein derartiges Vorgehen oft nicht möglich. Als Ausweg bietet sich oft die automatische Segmentierung mittels generischer Modelle an. Falls der Satzinhalt jedoch unbekannt ist, so ist mit der Segmentierung eine Erkennung der gesprochenen Wortfolge verbunden. Eine automatische Erkennung birgt jedoch immer das Risiko von Fehlerkennungen und damit Fehlsegmentierungen. Werden derartig fehlsegmentierte Daten zum unüberwachten Training der Hidden-Markov-Modelle verwandt, führt dies u.U. zu einem kompletten Fehltraining der Modelle und letztlich zu einem vollständigen Versagen des Systems.

Als “Ausweg” bietet sich die sprecherspezifische Anpassung der SI-Modelle. Bei der sogenannten Sprecheradaption (SA: engl.: ’speaker adaptive’) werden SI-Modelle so modifiziert, so dass sie für einen bestimmten Benutzer besser geeignet sind. Die entstehenden Modelle stellen einen Mittelweg zwischen SI- und SD-Modelle dar. Bei steigender Zahl von Adaptionsäußerungen nähern sich SA-Modelle der Performanz eines SD-Systems [Hua91]. Häufig muss jedoch von wenig Adaptionsäußerungen ausgegangen werden. Das Ziel ist es, mit den wenigen, verfügbaren Daten eine hochgradige Anpassung an den jeweiligen Anwender zu erreichen. In der Literatur wurden verschiedene grundlegende Algorithmen und Vorgehensweisen vorgeschlagen, um ausgehend von SI-Basismodellen auf ein SA-System überzugehen [Fur89, Gau92, Leg95, Kuh98]. Einige dieser Verfahren finden auch in dieser Arbeit Anwendung und wurden in Kapitel 2 diskutiert.

Im Gegensatz zum SD-Training kann bei einer Adaption nicht von ausreichend Datenmaterial ausgegangen werden, um jeden Systemparameter individuell zu schätzen. Bei Adaptionsalgorithmen wird daher versucht, durch zusätzliche Annahmen oder zusätzliches Wissen die Zahl der freien, zu schätzenden Parameter einzuschränken. Typisches Beispiel hierfür sind die Annahme einer gemeinsamen Transformation, wie sie beim MLLR-Ansatz (s. Kap. 2.4) eingeführt wird, oder die Bestimmung von einheitlichen Bewegungsrichtungen der Parame-

ter, wie etwa bei VFS [Tak95, Fab97]. In ähnlicher Form wird das Wissen über die sprecherabhängige Variation der Modellparameter auch beim Eigenvoice-Ansatz dazu benutzt, um den Freiheitsgrad der zu schätzenden Unbekannten zu begrenzen [Kuh98, Kuh99, Kuh00]. Ein gänzlich anderer Ansatz wird bei MAP verfolgt. Hier werden die Modellparameter als statistische Größen mit einer definierten Verteilungsfunktion angenommen[Gau92].

Die Gesamtzahl der freien Modellparameter übersteigt i.d.R. die Millionengrenze, wobei der Hauptanteil auf die Mittelpunkte der (Normal)Verteilungen sowie deren Kovarianzmatrizen entfällt. Tabelle 5.1 zeigt die Zahl der Parameter, die robust geschätzt werden müssen, für die beiden in dieser Arbeit primär verwendeten SI-Systeme.

Modelltyp	Monophon	Triphon
#Verteilungen	8500	35000
#Zustände	135	2500
	#Parameter	
Mittelpunkte	357000	1470000
Varianzen	357000	1470000
Mixturgewichte	8500	35000
Transitionen	405	fix
Gesamt	722000	2975000

Tab. 5.1: Parameterzahl der beiden verwendeten HMM-Basismodellsätze.

Ein Kritikpunkt ergibt sich jedoch bei den erläuterten Adaptionsverfahren. Nahezu alle der Trainingsverfahren benötigen zum Nachschätzen der Modellparameter (vgl. Abschnitt 2.3.1) eine phonetische Segmentierung der Adaptionsdaten. Wie bei den Ausführungen zum SD-Training angeführt, setzt die Segmentierung einer Trainingsäußerung - bei unbekannter Transkription - einen automatischen Erkennungsdurchlauf voraus. Das Erkennungsergebnis ist jedoch häufig fehlerbehaftet, was u.U. zu einer Fehladaption führen kann. In Tabelle 2.1 konnte dies für MLLR-, sowie MLED-adaptierte Modelle gezeigt werden.

Ein Kernziel dieser Arbeit war die Entwicklung und Untersuchung von Gruppierungsmethoden, die es ermöglichen Information über den Sprecher bereits bei der Modellbildung einzuarbeiten, um auf diese Weise sprecherähnlichere Modelle zur Spracherkennung generieren zu können. Diese sollen einerseits eine verbesserte Verschriftung als SI-Modelle erlauben und andererseits eine bereits optimierte Ausgangsbasis für eine nachfolgende Adaption bilden [Gal00, Joh98, Hec97, Kos94a, Kos94b, Pad98, Gao97]. Als Nebenbedingung ist jedoch hier die Forderung anzusehen, *keine phonetische Segmentierung der Adaptionsdaten* zu benötigen. Die Methoden sind primär darauf ausgerichtet, Ähnlichkeiten zwischen dem neuen Sprecher und den Sprechern, die in einer Trainingsdatenbank verfügbar sind, aufzudecken. Hintergedanke ist hierbei, das Trainingsmaterial deutlich effizienter und zielgerichteter zu nutzen.

Die Methoden, die im folgenden untersucht werden, konzentrieren sich auf das Konzept

der automatischen Sprechergruppierung (engl. 'speaker clustering'), sowie der automatischen Sprecherzuordnung in der Erkennungsphase. Die Idee hinter diesem Ansatz ist, statt einem einzigen SI-Modellsatz mehrere verschiedene HMM-Sätze zu trainieren. Diese können für charakteristische Sprecher oder für repräsentative Gruppen von Sprechern generiert werden. Die zwei Hauptprobleme, die sich bei diesem Ansatz stellen, sind einerseits die Auswahl des zu verwendenden Modellsets in der Erkennungsphase und andererseits das Auffinden oder Festlegen von charakteristischen Sprechergruppen. Letzteres sollte automatisch, d.h. ohne explizites Expertenwissen erfolgen.

Bei beiden Fragestellungen liegt die grundlegende Schwierigkeit implizit darin, geeignete Maße und Modelle zu finden, die robuste Aussagen über die Ähnlichkeit von Sprechern treffen. Erschwert wird dies dadurch, dass sich ein Sprecher im Prinzip durch seinen gesamten akustisch-phonetischen Artikulationsbereich definiert. Im Merkmalsraum wird für einen Sprecher dementsprechend ein weiter Bereich aufgespannt, der sich mit dem anderer Sprecher *überlappt*. Als Beispiel: ein /a/ von Sprecher A ist einem /a/ eines beliebigen Sprechers B sehr wahrscheinlich im Merkmalsraum näher gelegen als beispielsweise ein /s/ desselben Sprechers A. Abb. 5.1 veranschaulicht diesen Sachverhalt graphisch im 2-dimensionalen Raum.

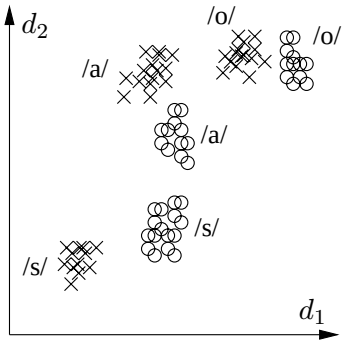

Abb. 5.1: Muster im Merkmalsraum für 2 Sprecher A (×) und B (○).

Die schematische Darstellung in Abb. 5.1 macht bereits zwei der untersuchten Problemstellungen deutlich. Aufgrund der Vielzahl der Mustervektoren und der daraus resultierenden Menge an Kombinationsmöglichkeiten ist eine Ähnlichkeitsbewertung auf Vektor-zu-Vektor Basis zu aufwendig. Daher müssen für jeden Sprecher geeignete Abstraktionen, d.h. Modelle gefunden werden. Die Eigenschaft, dass sich die Sprecher im Merkmalsraum überlappen wirft darüber hinaus die Fragestellung auf, wie Ähnlichkeit zu messen und zu bewerten ist. Von Interesse ist, ob es beispielsweise genügt, wenn 2 Sprecher sich in einem Laut ähnlich sind, oder ob in allen Lauten Übereinstimmung herrschen muss.

In der Literatur wurden bereits verschiedene Modellstrukturen zur Repräsentation von Sprechern vorgeschlagen. In dieser Arbeit wurden schwerpunktmäßig prototyp-basierte Modelle daraufhin untersucht, inwieweit diese eingesetzt werden können, um Ähnlichkeitsbeziehungen zu erfassen. Die untersuchten Modelle werden im folgenden Abschnitt 5.2 vorgestellt. Als Kriterium zur Beurteilung der Leistungsfähigkeit der Modelle wird die Sprecheridentifi-

kationsrate (IR) vorgeschlagen. Sie gibt Aufschluss darüber, wieviele Merkmalsvektoren dem korrekten Sprecher zugeordnet werden können. Die untersuchten Modelle, sowie die entwickelten Erweiterungen werden anhand dieses Kriteriums bewertet.

5.2 Sprechermodelle

5.2.1 Modellstrukturen

In diesem Abschnitt werden die Modellstrukturen erläutert, die in dieser Arbeit untersucht und erweitert wurden, um Sprecher bzw. deren akustische Eigenschaften zu beschreiben. Die Abstraktionen dienen als Ausgangsbasis, um Abstandsmaße zwischen Sprechern festzulegen.

- VQ (1 Modell pro Sprecher)
- GMM (1 Modell pro Sprecher)
- Phonemweise Modellierung mit HMMs (N_{Ph} Modelle pro Sprecher)
- Phonemgruppenweise Modellierung mit GMMs (N_I Modelle pro Sprecher)

Alle Strukturen setzen dabei auf eine prototypische Repräsentation der sprecherspezifischen Merkmalsvektoren. Bei VQ-Modellen kommen als Prototypen einfache Mittelpunktsvektoren und daraus resultierend Abstandsklassifikatoren zum Einsatz. Bei Gauss'schen-Mixturmodellen [Rey95] bzw. Hidden-Markov-Modellen handelt es sich um stochastische Modelle, die - in der implementierten Form - auf Gauss'sche Normalverteilungen zur Abstraktion der Merkmalsvektoren bauen. Die Ähnlichkeitsbewertung erfolgt dementsprechend durch Wahrscheinlichkeitsmaße. Die Modellstrukturen haben bereits für die Sprechererkennung gute Resultate gezeigt und sich gegenüber anderen Verfahren durchgesetzt. Im Anschluss folgt die Beschreibung der Strukturen sowie der entwickelten Erweiterungen.

5.2.1.1 VQ-Modelle

VQ-Modell (VQ, engl. "Vector Quantization") ist im Prinzip ein Codebuch mit K Prototypen. Als Prototypen werden die Mittelpunktsvektoren von Vektorballungen (Cluster) im Merkmalsraum gesucht. Diese können mittels automatischer Clusterverfahren, z.B. LBG, EMR, OC (s. Abschnitt 2.2) gefunden werden. Insbesondere bei geschlossenen Systemen (fixe Sprecherzahl) lassen sich die Prototypen durch diskriminative Verfahren weiter anpassen [He97, He99b, Nae01]. Als wesentlicher Unterschied zu den nachfolgend erläuterten statistischen Klassifikatoren kann gesehen werden, dass es sich bei einem VQ-Modell um eine Struktur zur Abstandsklassifikation handelt. Im allgemeinen wird der ungewichtete Euklidische Abstand verwendet. Der Abstand eines beliebigen Vektors zu dem Modell wird bestimmt durch den Mittelpunkt (von K Mittelpunkten), der dem Vektor am nächsten gelegen ist

$$d_m(\boldsymbol{x}_j) = \min_{k=1..K_m} \|\boldsymbol{x}_j - \boldsymbol{\mu}_{mk}\| \tag{5.1}$$

Bei der Anwendung als Modell zur Sprechercharakterisierung wird für jeden Sprecher ein eigene VQ-Repräsentation trainiert. Die Modelle können auch zur Zuordnung einer Sequenz

von (unbekannten) Mustervektoren zu einem Sprecher eingesetzt werden. Bei der Klassifikation wird jedoch die Entscheidung für eine Klasse anhand mehrerer Mustervektoren getroffen [He99a].

$$D_m(\boldsymbol{\mathcal{X}}) = \sum_{j=1}^{T} d_m(\boldsymbol{x}_j)\delta(\boldsymbol{x}_j) \tag{5.2}$$

$$\text{mit} \qquad \delta(\boldsymbol{x}_j) = \begin{cases} 1 & \text{sichere Sprache} \\ 0 & \text{sonst} \end{cases}$$

Zur Klassifikation werden nur diejenigen Frames herangezogen, die von einem unabhängigem Sprach-Pause Detektor [Beh95b] als Sprache gekennzeichnet wurden. In einer weiter restringierten Variante werden nur diejenigen Vektoren in Gl. 5.2 einbezogen, die eine ausreichende Konfidenzsicherheit aufweisen [Bes98].

5.2.1.2 Gauss'sches Mixturmodell (GMM)

Das Gauss'sche oder auch Generische Mixturmodell (GMM) [Rey95] ist ein stochastisches Modell, das die WDF der zu modellierenden, mehrdimensionalen Mustervektoren mittels *einer* gewichteten Überlagerung aus Gauss'schen Normalverteilungen nachbildet - wobei jedoch auch andere Prototypverteilungen (z.B. Laplace) möglich sind. In der von Reynolds in [Rey95] vorgeschlagenen Form erfolgt keine Unterteilung in Einheiten, weder in phonetischer (z.B. Phoneme), noch in zeitlicher Hinsicht (z.B. Zustände). Die gesamte Verteilung der Trainingsvektoren eines Sprechers wird durch eine einzelne, zeitlich unstrukturierte Verteilung wiedergegeben. Zeitliche Abhängigkeiten in der Abfolge der Merkmalsvektoren bleiben somit unberücksichtigt. Ein GMM kann als Spezialfall eines HMMs mit nur einem Zustand gesehen werden. Das Modell selbst ist eine Überlagerung aus K Gaussverteilungen, wobei jede mit einem Mixturkoeffizienten c_k gewichtet wird.

$$p(\boldsymbol{x}_j|m) = \sum_{k=1}^{K_m} c_{mk}\mathcal{N}(\boldsymbol{x}_j, \boldsymbol{\mu}_{mk}, \boldsymbol{\Sigma}_{mk}) \tag{5.3}$$

Diese Gleichung beschreibt die Wahrscheinlichkeit des Modells für *einen* Vektor. Zur Klassifikation von unbekannten Testdaten wird jedoch ebenfalls eine Sequenz von Merkmalsvektoren herangezogen [He99a]:

$$p(\boldsymbol{\mathcal{X}}|m) = \prod_{j=1}^{T} p(\boldsymbol{x}_j|m) \tag{5.4}$$

bzw.

$$S_m(\boldsymbol{\mathcal{X}}) = \sum_{j=1}^{T} \log p(\boldsymbol{x}_j|m) \tag{5.5}$$

Bei Einsatz dieser Modelle zur Klassifikation werden, um zu einer Entscheidung für ein Modell $\hat{m}$ zu gelangen, i.d.R. alle K_S Modelle parallel bewertet und dasjenige ausgewählt, das die höchste Bewertung liefert:

$$\hat{m} = \arg \max_{m=1..K_S} S_m(\boldsymbol{\mathcal{X}}) \tag{5.6}$$

Zur Initialisierung der Modelle wurden in dieser Arbeit die in Kapitel 2 erläuterten Clusteralgorithmen eingesetzt. Die nachfolgende Optimierung der Prototypen wurde anhand der ebenfalls in diesem Kapitel beschriebenen Trainingsverfahren untersucht.

5.2.1.3 Phonemweise Modellierung (HMMs)

Die Verwendung von Hidden-Markov-Modellen stellt die zur Spracherkennung ähnlichste Form der Modellierung zum Zwecke der Sprechererkennung dar. Jeder Sprecher wird durch $N_M = N_{Ph}$ HMMs (s. 1.2) repräsentiert, wobei ein jedes der Modelle den akustischen Merkmalsraum eines Phonems des Sprechers nachbildet (Monophon-HMMs). Im Gegensatz zur Spracherkennung, bei der die genauen Musterverläufe von Bedeutung sind, ist bei der Sprechererkennung primär deren generelle Lage im Merkmalsraum von Interesse. Dies zeigt sich in der Wahl der Parameterzahl der HMMs. Werden zur Spracherkennung sehr viele Normalverteilungen in den HMM-Zuständen benötigt (vgl. Tab. 5.1), genügen zur Sprechererkennung vergleichsweise wenige Dichten, meist nur 1..3 je Zustand. Die geringe Zahl hat Auswirkungen auf die Parameterwerte der Verteilungen. Insbesondere die Varianzen weisen deutlich höhere Werte auf. Die Gesamtzahl der freien Parameter ist somit vergleichbar mit der bei Sprecher-GMMs. Bei 40 Phonem-HMMs mit jeweils 3 Zuständen und 1..3 Verteilungen je Zustand ergibt sich die Zahl der Verteilungen zu $N_{Bf} \approx 120 \ldots 360$.

Der Einsatz von Hidden-Markov-Modellen zur Sprechermodellierung stellt einen gewissem Gegensatz zum Einsatz von GMMs dar. Bei letzteren wird der *gesamte* "akustische" Merkmalsraum durch eine einzelne WDF in Form einer Mixturverteilung repräsentiert. Bei HMMs hingegen erfolgt einerseits eine Einteilung des Merkmale in Laute bzw. Lautklassen, die individuell repräsentiert werden. Andererseits wird durch die HMM-Zustände noch in gewissem Umfang die Repräsentation der zeitlichen Laut-Strukturierung ermöglicht.

Die Abarbeitung von Sprecher-HMMs zur Bewertung von sprachlichen Äußerungen kann als freilaufende Phonemerkennung interpretiert werden. Die Entfaltung der Modelle bei der Erkennung erfolgt mittels des Viterbi-Algorithmus ([For73], s. Kap. 1.2.3). Im Gegensatz zur Phonemerkennung, bei der die Phonem- bzw. Zustandseinteilung im Vordergrund steht, ist zur Sprecherklassifikation nur der jeweils erzielte Gesamtscore S_m^{HMM} der Sprecher von Interesse. Die Entscheidung für einen Sprecher $\hat{m}$ kann analog zu Gl. 5.6 erfolgen.

5.2.1.4 Phonemgruppenweise Modellierung: Parallele GMMs

Die Sprechermodellierung anhand von Phonemgruppen stellt eine Verallgemeinerung des phonemweisen Ansatzes dar, indem einzelne Laute zu Lautklassen zusammengefasst werden. Allerdings sind HMMs zur Modellierung dieser Klassenstrukturierung eher ungeeignet. Aufgrund der unterschiedlichen Zeitverläufe der zusammengefassten Einheiten ist die Zuordnung zwischen HMM-Zuständen und stationären Bereichen nicht mehr unmittelbar gegeben. Aus diesem Grund werden in dieser Arbeit GMM-Modelle zur Gruppen-Repräsentation eingesetzt. Bei Verwendung von GMMs hat der klassenbasierte Ansatz Auswirkungen auf die Trennung

der Modelle. Bei GMMs wird *eine* Mixturverteilung zur Repräsentation der sprecherspezifischen Merkmalsvektoren angenommen [Rey95], d.h. es wird keine Einteilung in phonetische Klassen vorausgesetzt. Diese kann jedoch zusätzlich eingeführt werden, wie in [Fal01a] gezeigt werden konnte.

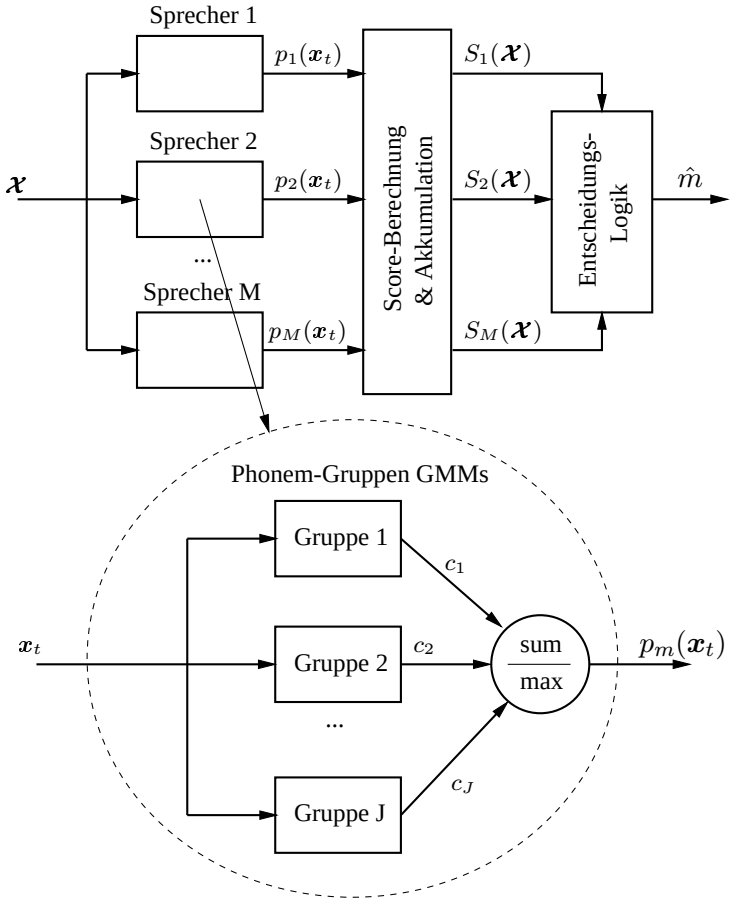

Abb. 5.2: Parallele Gruppen-GMMs für $J = N_I$ Phonemklassen für $M = K_S$ Sprecher.

Die Merkmalsvektormenge wird, analog zur phonemweisen Modellierung, in die phonetischen Klassen zerlegt. Jede der Phonemklassen wird durch eine eigene WDF=GMM wiedergegeben und kann folglich individuell trainiert werden. Für die Entfaltung der Modelle während der Erkennung ergeben sich jedoch zwei prinzipielle Alternativen. Einerseits können die Klassen-GMMs eines Sprechers in jedem Frame strikt separat bewertet werden.

$$p(\boldsymbol{x}_j|m) = \max_{N_I} p(\boldsymbol{x}_j|m,i) \tag{5.7}$$

Andererseits stellt jedes Teil-GMM immer einen Ausschnitt aus dem phonetischen Gesamtmerkmalsraum der globalen GMM-Modellierung dar. Alle zusammen bilden das Gesamt-GMM.

$$p(\boldsymbol{x}_j|m) = \sum_{i=1}^{N_I} p(\boldsymbol{x}_j|m,i) \tag{5.8}$$

Nach Gl. 5.8 können die Verteilungen der Teil-GMMs zusammengefasst und strukturell in ein GMM integriert werden. Damit wäre also nur das Schätzen der klassenweisen Verteilungsparameter individuell erfolgt - der Erkennereinsatz könnte jedoch mit einem einzelnen, gemeinsamen Modell erfolgen. Auswirkungen hat dies auf die Mixturkoeffizienten der einzelnen Verteilungen. Wird ein Phonemklassen-GMM i eines Sprechers m separat trainiert, so gilt für dessen Mixturkoeffizienten c_{mik}:

$$\sum_{k=1}^{K_{mi}} c_{mik} = 1 \tag{5.9}$$

Werden die Verteilungen der N_I Teilmodelle in einem Modell zusammengefasst, so muss für deren Koeffizienten diese Bedingung äquivalent gelten.

$$\sum_{k^*=1}^{K_m} c^*_{mk^*} = \sum_{i=1}^{N_I} \sum_{k=1}^{K_{mi}} c_{mik} = 1 \tag{5.10}$$

mit $K_m = \sum_{i=1}^{N_I} K_{mi}$. Wie die Gleichung bereits andeutet, bietet sich an dieser Stelle die Einführung einer phonetischen Gewichtung c_i an. Der Beitrag der verschiedenen Phonemklassen zur Gesamtwahrscheinlichkeit wird hierdurch unterschiedlich bewertet $c^*_{mk^*} = c_i c_{mik}$. Damit ändert sich Gl. 5.10 zu

$$\sum_{i=1}^{N_I} c_i \sum_{k=1}^{K_{mi}} c_{mik} = \sum_{i=1}^{N_I} c_i = 1 \qquad \forall m \tag{5.11}$$

Da die Koeffizienten c_{mik} nach Gl. 5.9 bereits klassenweise normiert sind, können die Koeffizienten c_i nach Gl. 5.11 vorteilhafterweise sowohl unabhängig von den Mixturgewichten c_{mik}, als auch *unabhängig von den Sprechern* eingestellt werden. Als Bewertungsmaß bietet sich neben der Gleichverteilung $c_i = \frac{1}{N_I}$, insbesondere die Apriori-Wahrscheinlichkeit $p(i)$ der Klassen an. Diese lässt sich - sprecherunabhängig - aus den Trainingsdaten schätzen.

$$p(i) \approx \frac{N_i^{Phonem}}{\sum_{i=1}^{N_I} N_i^{Phonem}} \quad \text{bzw.} \quad p(i) \approx \frac{N_i^{Frame}}{\sum_{i=1}^{N_I} N_i^{Frame}} \tag{5.12}$$

Die Abschätzung von $p(i)$ kann entweder anhand der phonemweisen Auftretenshäufigkeit N_i^{Phonem} der Phoneme einer Klasse i erfolgen, oder anhand der Frame-weisen Auftretenshäufigkeit N_i^{Frame}. Der Gebrauch der Apriori-Wahrscheinlichkeit für die Gewichtung der Phonemklassen berücksichtigt jedoch nur deren quantitatives Auftreten - nicht jedoch deren qualitativen Beitrag zur Sprechertrennung [Eat94, Par94, Auc99].

Bei qualitativer Betrachtungsweise sollten die Gewichte c_i so eingestellt werden, dass die Trennbarkeit bzw. Identifizierbarkeit der Sprecher ein Maximum annimmt. Die Trennbarkeit kann näherungsweise anhand der Score-Distanz [Fal01a] angegeben werden zu:

$$\Delta L(\boldsymbol{\mathcal{X}}_r, \boldsymbol{c}) = L(\boldsymbol{\mathcal{X}}_r | m, \boldsymbol{c}) - L(\boldsymbol{\mathcal{X}}_r | \bar{m}, \boldsymbol{c}) \tag{5.13}$$

Die log-Likelihood Differenz gibt den Abstand des korrekten Modells m zu einem konkurrierenden Sprecher $\bar{m}$ an. Dies kann sowohl der am stärksten konkurrierende Sprecher sein,

als auch ein generisches SI-Modell. Die Likelihood-Funktionen hängen nichtlinear von $\boldsymbol{c}$ ab, daher lässt sich eine Optimierung bzgl. $\boldsymbol{c}$ durchführen. Das Gesamttrennbarkeitsmaß über alle R Trainingsäußerungen ergibt sich zu:

$$\Delta L(\boldsymbol{c}) = \sum_{r=1}^{N_R} L(\boldsymbol{\mathcal{X}}_r, \boldsymbol{c}) \tag{5.14}$$

Die Summation läuft hier über alle Äußerungen $\boldsymbol{\mathcal{X}}_r$ der Trainingsdaten. Die Optimierung kann mittels eines Gradientenverfahrens erfolgen, mit dem Ziel einer Maximierung des Abstands ΔL:

$$c_i(k+1) = c_i(k) + \epsilon \frac{\partial}{\partial c_i} \Delta L(k) \tag{5.15}$$

Die partielle Ableitung bzgl. eines Gewichtungskoeffizienten c_i ergibt sich zu:

$$\frac{\partial}{\partial c_i} \Delta L(\boldsymbol{c}) = \sum_{r=1}^{N_R} \frac{\partial}{\partial c_i} \Delta L(\boldsymbol{\mathcal{X}}_r, \boldsymbol{c}) \tag{5.16}$$

Die Likelihood einer Äußerung folgt aus dem Produkt der Frame-weisen Wahrscheinlichkeiten:

$$L(\boldsymbol{\mathcal{X}}_r, \boldsymbol{c}) = \log \prod_{t=1}^{T_r} p(\boldsymbol{x}_{rt}|m) = \sum_{t=1}^{T_r} \log p(\boldsymbol{x}_{rt}|m) \tag{5.17}$$

Eingesetzt in Gleichung 5.13 bzw. 5.16, und unter Ausnutzung der Kettenregel für Ableitungen ergibt sich somit:

$$\frac{\partial}{\partial c_i} \Delta L(\boldsymbol{\mathcal{X}}_r, \boldsymbol{c}) = \sum_{t=1}^{T_r} \frac{1}{p(\boldsymbol{x}_{rt}|m)} \frac{\partial}{\partial c_i} p(\boldsymbol{x}_{rt}|m) - \frac{1}{p(\boldsymbol{x}_{rt}|\bar{m})} \frac{\partial}{\partial c_i} p(\boldsymbol{x}_{rt}|\bar{m}) \tag{5.18}$$

mit:

$$\begin{aligned} \frac{\partial}{\partial c_i} p(\boldsymbol{x}_{rt}|m) &= \frac{\partial}{\partial c_i} \sum_{i=1}^{N_I} p(\boldsymbol{x}_{rt}|m,i) = \frac{\partial}{\partial c_i} \sum_{i=1}^{N_I} c_i \sum_{k=1}^{K_i} c_{mik} \mathcal{N}(\boldsymbol{x}_{rt}, \boldsymbol{\mu}_{mik}, \boldsymbol{\Sigma}_{mik}) = \\ &= \sum_{k=1}^{K_i} c_{mik} \mathcal{N}(\boldsymbol{x}_{rt}, \boldsymbol{\mu}_{mik}, \boldsymbol{\Sigma}_{mik}) = \frac{1}{c_i} p(\boldsymbol{x}_{rt}|m,i) \end{aligned} \tag{5.19}$$

Definiert man:

$$\gamma_m(\boldsymbol{x}_{rt}) = \frac{1}{c_i} \frac{p(\boldsymbol{x}_{rt}|m,i)}{p(\boldsymbol{x}_{rt}|m)} \tag{5.20}$$

so reduziert sich der Gradient zu:

$$\frac{\partial}{\partial c_i} \Delta L(\boldsymbol{\mathcal{X}}_r, \boldsymbol{c}) = \sum_{t=1}^{T_r} (\gamma_m(\boldsymbol{x}_{rt}) - \gamma_{\bar{m}}(\boldsymbol{x}_{rt})) \tag{5.21}$$

Der Summationsterm in diesem Teilgradienten kann als die Differenz aus den 2 normierten Aposteriori-Wahrscheinlichkeiten für die Phonemklasse i interpretiert werden. Ist der Rückschluss beim korrekten Sprechermodell m größer als beim Konkurrenzmodell, so ergibt sich

ein positiver Beitrag zum Gradienten - das Gewicht dieser Phonemklasse wird erhöht. Ist hingegen der Rückschluss beim jeweils konkurrierenden Modell höher, so wird daraufhin das Gewicht entsprechend verringert.

Kritisch sind an diesem Gradientenansatz die sich ergebenden Nebenbedingungen, die für die Mixturgewichte eingehalten werden müssen (Gl. 5.11). Insbesondere dürfen diese Faktoren nicht negativ werden, d.h. $c_i > 0 \quad \forall i = 1 \ldots N_I$. Die Bedingung lässt sich durch die Einführung eines minimalen Schwellwerts einhalten, auf den c_i gegebenenfalls zurückgesetzt wird. Die Summenbedingung in Gl. 5.11 kann durch eine Normierung nach jedem Iterationsschritt k gewährleistet werden.

$$\tilde{c}_i(k+1) = c_i(k) + \epsilon \frac{\partial}{\partial c_i} \Delta L(k) \tag{5.22}$$

$$c_i(k+1) = \frac{\tilde{c}_i(k+1)}{\sum_{j=1}^{N_I} \tilde{c}_j(k+1)} \tag{5.23}$$

Der Vorteil dieses Ansatzes gegenüber einem rein diskriminativen Trainingsverfahren kann darin gesehen werden, dass die Sprechermodelle unabhängig voneinander trainiert werden können. Aufbauend auf den fertig trainierten Modellen lassen sich davon wiederum unabhängig im Anschluss die phonetischen Gewichte ermitteln. Die Einstellung erfolgt mit dem Ziel einer verbesserten Trennbarkeit der Sprecher. Bei rein diskriminativen Verfahren zum Modelltraining werden normalerweise die Parameter (Mittelwerte,Varianzen) der sprecherspezifischen Modelle geschätzt. Dies geschieht unter Berücksichtigung der zum Trainingszeitpunkt vorliegenden 'Konkurrenzsituation' der Trainingssprecher. Bei neu hinzukommenden Sprechern könnte sich diese stark ändern. Für sprecheroffene Systeme stellt dies einen deutlichen Nachteil dar.

Beim vorgestellten Algorithmus werden die phonetischen Gewichte zwar ebenfalls diskriminativ eingestellt, jedoch handelt es sich tendenziell um sprecherunabhängige, globale Parameter. Bei einer genügend großen Sprecherstichprobe kann davon ausgegangen werden, dass die Parameter allgemeingültig optimiert werden. Tab. 5.2 zeigt die ermittelten Gewichtungsfaktoren für einige typische Phonemklassen, bei insgesamt 10 Klassen. Als Konkurrenzmodell $\bar{m}$ wurden ein generisches, sprecherunabhängiges GMM ('Generisch') sowie das Modell des am stärksten konkurrierenden Sprechers ('Top-1') eingesetzt.

Im folgenden Abschnitt wird anhand von Identifikationsexperimenten eine Bewertung der unterschiedlichen Modellstrukturen vorgenommen. Bei diesen Versuchen wurde auch der Einfluss der optimierten Gewichtungsfaktoren untersucht.

5.2.2 Aufbau des Klassifikationssystems

Die untersuchten Modellstrukturen kommen in zweierlei Hinsicht zum Einsatz. Auf Basis der durch die Modelle möglichen Abstandsmaße lassen sich Sprecher zu Gruppen zusammenfassen. Für diese Gruppen können spezialisierte akustische Erkenner-HMMs trainiert werden [Ima91]. Die Zielaufgabe ist allerdings darin zu sehen, das Erkennungssystem an den jeweiligen aktuellen Anwender anzupassen, der dem System zum Zeitpunkt der Modellerstellung

Klasse	Phoneme	Top-1	Generisch
Nasale 1	/m/	0.049	0.072
Nasale 2	/n/ /N/	0.127	0.152
Frikative	/s/ /S/ /z/ /Z/ /f/ /v/ /C/ /x/ /j/	0.243	0.160
Liquide	/l/ /r/	0.092	0.078
Plosive	/t/ /k/ /d/ /b/ /p/ /g/	0.268	0.147
Vokale 1	/E:/ /e:/ /E/	0.034	0.074

Tab. 5.2: Zusammensetzung der phonetischen Klassen bei Verwendung von 10 Klassen zusammen mit dem ermittelten phonetischen Gewicht bei unterschiedlichem Konkurrenzmodell.

noch nicht bekannt ist. Zur Laufzeit muss also eine rasche, rechenzeittechnisch unaufwendige Zuordnung des Sprechers zu einem vorhandenen Sprecher- oder Sprechergruppenmodell erfolgen. Diese Ähnlichkeitszuordnung sollte innerhalb der ersten Sekunden erfolgen. Realisiert wurde diese mittels des in Abb. 5.3 dargestellten Klassifikationssystems, das im Prinzip einer direkten Umsetzung von Gl. 5.6 entspricht.

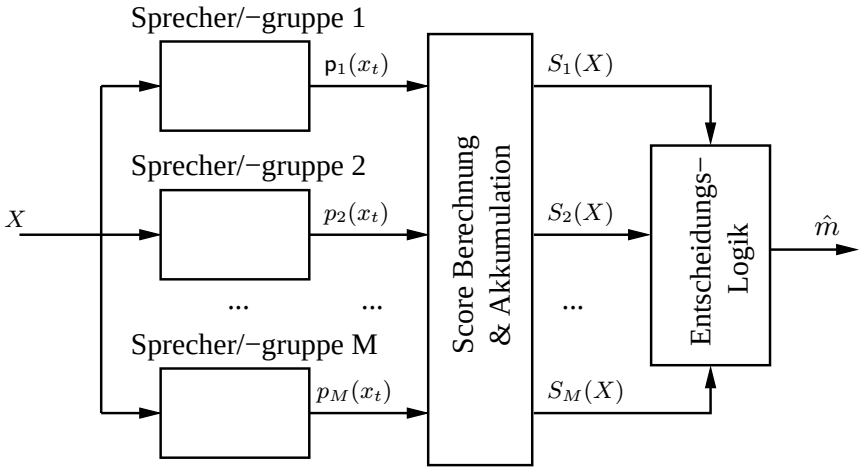

Abb. 5.3: Aufbau des implementierten Klassifikators zur Online- Sprecherzuordnung.

Die Modellierung eines Einzelsprechers stellt einen Spezialfall eines Sprechergruppenmodells dar, bei dem die "Gruppe" nur aus einem Individuum besteht. Soll die Auswahl allerdings unter allen zur Verfügung stehenden Sprechermodellen erfolgen, so müssen effiziente Pruningstrategien eingesetzt werden, um den Rechenaufwand gering zu halten. Eine fixe Pruningschwelle ist zu diesem Zweck sehr effektiv, da im Gegensatz zur Spracherkennung keine beliebigen Modellabfolgen [Pla95] bewertet werden müssen, sondern nur jeweils ein einzelnes Modell berechnet wird. Es erfolgt keine erneute Expansion des Suchraums während des Suchvorgangs. Dieser kann durch fortgesetztes Pruning von Sprechern also nur kleiner werden. Experimente zur Sprecherauswahl aus den 613 Verbmobil Sprechern zeigen, dass bei GMM-Modellen die Suche, ohne Verlust an Optimalität, innerhalb von 50 Frames auf unter 50 Sprecher reduziert werden kann.

5.2.3 Bewertung der Modelle

Die aus den Merkmalsvektoren eines Sprechers geschätzten parametrischen Modelle stellen eine Abstraktion eines Sprechers dar. Ziel ist es, mittels der gefundenen Abstraktionen Aussagen über Ähnlichkeiten zwischen Sprechern des Trainingskorpus und neuen, unbekannten Sprechern bzw. deren Merkmalsvektoren zu finden. Jede der oben angeführten Modellstrukturen ist darauf ausgelegt eine Bewertung eines Mustervektors zu liefern, etwa in Form eines Abstandsmaßes (VQ) oder in Form einer Wahrscheinlichkeit (HMM,GMM). Eine Schwierigkeit hierbei ist dieses Ähnlichkeitsmaß qualitativ einzuschätzen.

Im folgenden wurde versucht die Qualitätseinstufung anhand von Sprecheridentifikationsexperimenten vorzunehmen: Die Qualität eines Modells kann indirekt daran gemessen werden, wie gut es in der Lage ist unbekannte, *klasseneigene* Mustervektoren zu klassifizieren. Je weniger ein Modell in der Lage ist eigene Vektoren korrekt zu klassifizieren, desto weniger verlässlich ist auch dessen Ähnlichkeitsaussage bezüglich klassenfremder Mustervektoren. Die Identifikationsrate (IR) selbst ist abhängig von der Anzahl der bei der Erkennung konkurrierenden Sprecher. Das Ziel ist allerdings ein Vergleich von Modellstrukturen, die alle mit dem selben Identifizierungsproblem konfrontiert werden.

Die folgenden Tabellen zeigen die Identifikationsleistung der verschiedenen Modellstrukturen in Abhängigkeit von der Zahl der verwendeten Parameter, der eingesetzten Vorverarbeitung, sowie der Zahl der bewerteten Mustervektoren. Die Experimente wurden auf Basis der Xval96-Crossvalidierungsdaten (74 Sprecher) unter Verwendung der in Abb. 5.3 bzw. 5.2 gezeigten Klassifikationssysteme durchgeführt.

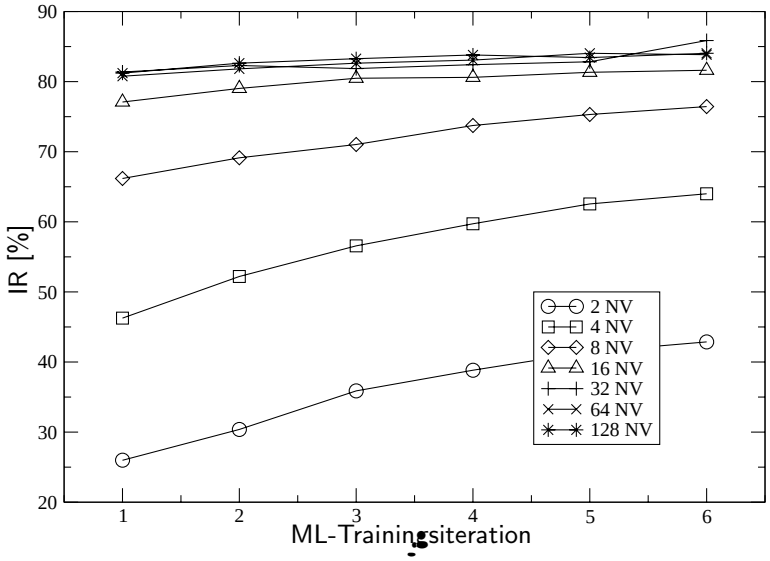

Abb. 5.4: Sprecheridentifizierungsrate (IR) von inkrementell, ML-trainierten GMMs (74 Sprecher, Merkmalsvektor MFCC12, 1s Test) zeigt Sättigung bei ca. 84% und einer GMM-Größe von 32 NV.

Die Kurven in Abb. 5.4 beschreiben die Abhängigkeit der IR von der Anzahl der Normalverteilungen bei ML-trainierten GMMs. Bei der gegebenen MFCC12-Vorverarbeitung ergibt sich eine Sättigung der IR bei etwa 84%. Diese wird bereits mit ca. 32 Verteilungen erreicht.

Tab. 5.4 zeigt die Sprecherzuordnungsrate bei Bewertung eines jeden einzelnen Merkmalsvektors. Auffallend ist, dass insbesondere die Nasallaute /m/, /n/ und /N/ hohe sprecherspezifische Information tragen. Je nach eingesetzter Vorverarbeitung können bei diesen Lauten ca. 30% der Merkmalsvektoren dem korrekten Sprecher (von 74 Sprechern) zugeordnet werden. Immerhin noch bis zu 20% der Merkmalsvektoren können bei den Vokal(klassen) /a:/ und /e:/, sowie den Frikativlauten korrekt zugeordnet werden. Insbesondere bei den Plosivlauten liegt diese Rate deutlich niedriger. Vergleicht man in Tab. 5.4 die Erkennungsraten zwischen den verschiedenen Vorverarbeitungen, so wird deutlich, dass die Hinzunahme der Deltakoeffizienten bei vielen Lauten eine Steigerung der IR ermöglicht. Die weitere Hinzunahme der DeltaDelta-Koeffizienten lässt zwar bei manchen Lauten eine geringfügige weitere Verbesserung zu, führt aber im Gegenzug bei einigen zu einer Verschlechterung. In Tab. 5.4 liegt die mittlere IR bei ca. 20%.

Verlängert man die Testsequenz auf 1s (s. Tab. 5.3), so kann hierdurch die IR auf etwa 85% gesteigert werden. Allerdings bleibt auch das Trainingsverfahren nicht ohne Auswirkung: speziell der reine ML-Ansatz ist tendenziell nachteilig, wenn das verfügbare Trainingsmaterial der Sprecher nicht phonetisch ausgewogen ist. Dies kann jedoch durch die strukturellen Modifikationen aus Abschnitt 5.2.1.4 ('mGMM') oder auch durch optimierte Cluster- ('EMR','OC') oder Trainingsverfahren ('MAP') kompensiert werden.

Modell	Train.Verf.	Anmerkung	N_{Bf}	IR (1 Sekunde)
GMM	ML		32	85.9
	ML		64	84.0
	ML		128	84.0
	MAP		64	87.5
	MAP	nur $\boldsymbol{\mu}$	64	88.1
	EMR		64	87.4
	EMR	+ML	64	86.0
	EMR		128	89.4
	OC		55	87.8
VQ	EMR		64	85.0
mGMM(10)	MAP	ungewichtet, max	64	88.8
		apriori, max	64	89.8
		top-1, max	64	89.3
		generisch, max	64	89.5
		ungewichtet, sum	64	88.7
		apriori, sum	64	89.5
		top-1, sum	64	89.4
		generisch, sum	64	89.5

Tab. 5.3: Sprecheridentifikationsrate bei unterschiedlichen Trainingsverfahren und Modellstrukturen für Testabschnitte der Länge 1 s und MFCC12-Vorverarbeitung.

Phonem	#Frames	IR [%]		
		MFCC12	MFCC24	MFCC36
n	27749	30.38	37.05	34.68
s	23571	17.04	23.32	20.87
t	15469	8.84	12.59	12.68
m	14599	28.97	33.50	30.58
a:	10770	21.58	25.85	23.70
aI	10363	20.52	27.60	26.35
f	10246	9.94	12.86	12.04
6	10119	17.33	18.63	18.48
a	8451	18.55	21.25	22.53
C	7793	13.10	18.34	15.24
d	7088	8.11	16.03	16.27
i:	6838	14.20	17.23	15.66
Q	6788	6.44	8.15	7.12
@	6507	14.88	17.77	17.84
I	6375	12.25	15.95	18.24
z	5867	15.36	20.91	18.22
e:	5512	17.53	20.46	18.72
k	5438	5.19	9.10	9.65
v	5061	7.03	10.69	9.50
E:	4960	20.24	25.83	22.60
l	4830	8.49	11.47	10.50
x	4449	9.22	12.83	12.34
E	4058	16.34	16.34	16.78
b	3668	9.21	14.89	14.29
S	3545	15.49	18.98	16.76
u:	3515	10.41	12.89	11.47
O	3437	15.13	16.53	16.38
o:	3397	11.60	11.78	11.01
r	3370	7.42	8.96	7.72
aU	3276	14.84	17.70	15.90
g	3239	6.95	10.65	11.45
U	3199	9.69	13.63	13.22
j	2852	9.85	12.10	11.22
p	2613	7.12	10.87	11.21
N	2550	23.76	28.78	25.76
h	1933	10.45	10.29	9.31
Y	1017	9.34	11.41	12.78
OY	1016	19.59	22.74	19.59

Tab. 5.4: Sprechererkennungsraten (74 Sprecher-GMMs mit je 64 NV) der einzelnen Phoneme auf Frameebene bei unterschiedlicher Zusammensetzung des Merkmalsvektors.

5.2.4 Auswirkung der Sprechgeschwindigkeit auf die Distanzberechnung

Die Untersuchungen in Abschnitt 3.4.1 haben gezeigt, dass bei HMM-Modellen zur Spracherkennung die mittleren HMM-Scores hochgradig von der gerade vorliegenden Sprechgeschwindigkeit beeinflusst sind und mit steigender LSR sinken. Die Korrelation erreicht hier Werte bis zu 0.65. Anders als bei der Modellierung zur Spracherkennung interessiert bei der Sprechererkennung mehr die allgemeine, sprecherspezifische Lage der Merkmalsvektoren im Merkmalsraum - der genaue Verlauf bzw. die Reproduzierbarkeit des Musterverlaufs von Äußerungen ist von untergeordneter Bedeutung. Aus diesem Grund werden bei der Modellierung, wie in den vorangegangenen Experimenten gezeigt wurde, tendenziell sehr wenige Gaussprototypen je Sprechermodell(satz) eingesetzt. In diesem Abschnitt wird untersucht, welche Auswirkungen Sprechgeschwindigkeit auf die Sprecherrepräsentation und damit auf die Sprecherdistanzberechnung hat.

Analog zu Abb. 3.10 ist in Abb. 5.5 der Score- sowie der LAS-Verlauf für die Beispieläußerung (wiederum negiert) dargestellt. Die LAS-Berechnung erfolgte für ein generisches, sprecherunabhängiges GMM-Modell (MFCC36, 64 NV, ML-trainiert).

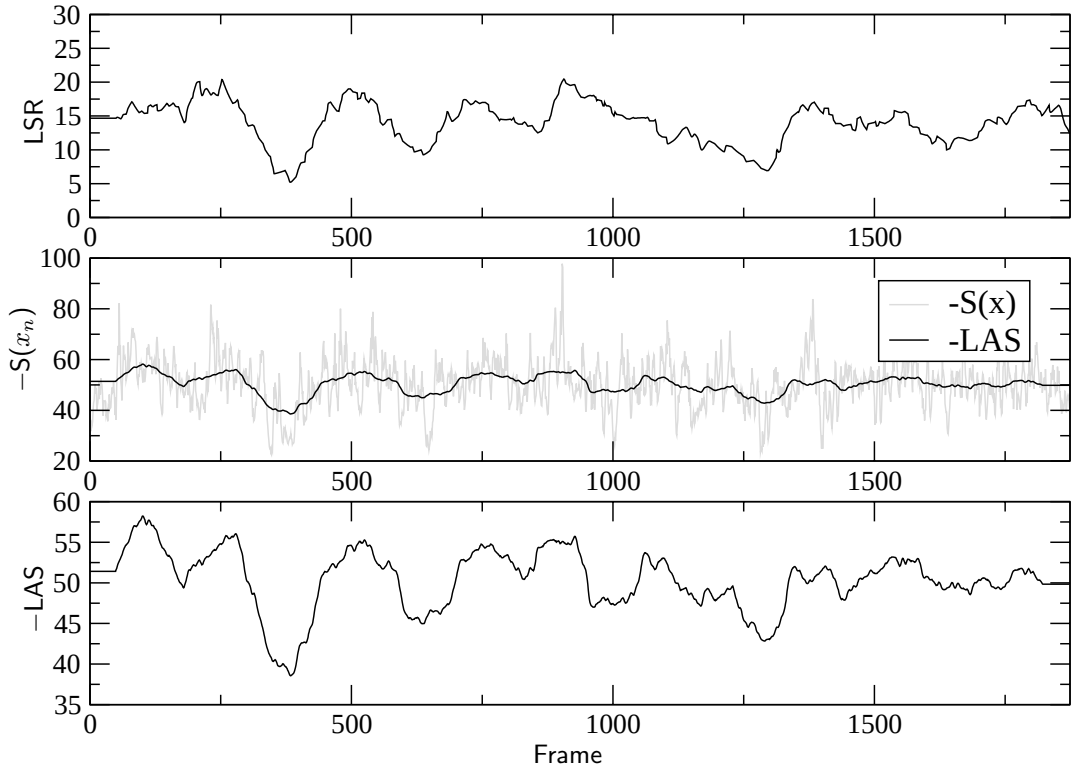

Abb. 5.5: Verlauf von LSR (oben), −Score und −LAS (unten) des Beispielturns 'g091a000' unter Verwendung eines generischen SI-GMMs.

Das generische SI-Modell, das aus den Daten der 613 Verbmobil-Sprecher trainiert wurde, und das auch eine Ausgangsbasis für die Generierung sprecherspezifischer Modelle bildet, stellt das Äquivalent zu den SI-Spracherkennungs-HMMs aus Kap. 3.4.1 dar. Interessanterweise zeigt der LAS-Verlauf in Abb. 5.5 einen prinzipiell analogen Verlauf, obwohl die Model-

lierung aufgrund der reduzierten Basisfunktionszahl deutlich "allgemeiner" sein sollte. Um die Abhängigkeit weiter aufzuschlüsseln wurde anhand von CD1 des Verbmobil Datenmaterials das LAS-Verhalten untersucht. Für verschiedene GMM- bzw. VQ-Modellgrößen wurde bei unterschiedlicher Zusammensetzung der Merkmalsvektoren der jeweilige Korrelationskoeffizient zwischen LAS und LSR ermittelt. Die resultierenden Ergebnisse sind in Tabelle 5.5 aufgeführt. Im Falle von VQ-Modellen wurde die Definition von LAS leicht abgeändert und der dynamische Mittelwert über die frameweisen Vektor-zu-Modell Distanzen (LAD, engl. "Local Average Distance") berechnet. Die Vektordistanz ergibt sich gemäß Gl. 5.1.

$$LAD(n) = \frac{1}{N_F} \sum_{j=n-\frac{N_F}{2}}^{n+\frac{N_F}{2}-1} d(\boldsymbol{x}_j) = \frac{1}{N_F} \sum_{j=n-\frac{N_F}{2}}^{n+\frac{N_F}{2}-1} \min_k \; d_k(\boldsymbol{x}_j) \tag{5.24}$$

Bei LAD handelt es sich im Gegensatz zu LAS jedoch um ein Distanzmaß. Dementsprechend ist ein niedrigerer LAD-Wert als "besser" anzusehen.

Modelltyp	Trainingsalg.	N_{Bf}	MFCC12	MFCC36
VQ	EMR	16	-0.028	0.507
	EMR	64	-0.022	0.520
GMM	MAP	16	0.034	-0.588
	MAP	64	0.029	-0.605

Tab. 5.5: Korrelationskoeffizient ρ_{las} (bzw. ρ_{lad} für VQ) zwischen Scoremittelwert und LSR für GMM und VQ-Modelle bei unterschiedlichem Merkmalsvektor.

Ähnlich wie bei den Untersuchungen in Abschnitt 3.4.1 ergibt sich auch hier ein starker Anstieg der Korrelation bei der Hinzunahme von Ableitungskoeffizienten in den Merkmalsvektor. Bei den rein statischen Merkmalen scheint der Einfluss der Sprechgeschwindigkeit vernachlässigbar.

In Abschnitt 3.4.3.3 wurde weiterhin gezeigt, dass der mittlere Scoreabstand LC_{T1} bzw. LC_{mean} zwischen dem zum jeweiligen Zeitpunkt korrekten Phonemmodell und konkurrierenden Phonemmodellen mit steigender lokaler Sprechgeschwindigkeit abnimmt. Die korrekten bzw. konkurrierenden Merkmalsvektoren, die in die Berechnung der beiden Maße eingehen, stammen stets vom selben Sprecher. Daher soll an dieser Stelle untersucht werden, inwieweit die Unterscheidbarkeit *zwischen Sprechern* durch die Sprechgeschwindigkeit beeinflusst wird. Zu diesem Zweck wurden für die in Verbmobil CD1 enthaltenen Sprecher individuelle GMM-Modelle trainiert. Das Training erfolgte mittels 3 Iterationen des MAP-Algorithmus, wobei als Ausgangsmodell das generische SI-Modell des vorangegangenen Experiments diente. Das SI-GMM, das als allgemeines Sprechermodell interpretiert werden kann, und demnach als "mittleres" Modell für beliebige Sprecher stehen kann, wurde als Konkurrenzmodell verwendet. Die LC_{mean}-Definition geht daher über in:

$$LC_{GMM}(n) = \frac{1}{N_F} \sum_{j=n-\frac{N_F}{2}}^{n+\frac{N_F}{2}-1} (S_{\hat{m}}(\boldsymbol{x}_j) - S_{SI}(\boldsymbol{x}_j)) \tag{5.25}$$

Äquivalent ergibt sich das lokale Konfidenzmaß LC_{VQ} zu:

$$LC_{VQ}(n) = \frac{1}{N_F} \sum_{j=n-\frac{N_F}{2}}^{n+\frac{N_F}{2}-1} (d_{\hat{m}}(\boldsymbol{x}_j) - d_{SI}(\boldsymbol{x}_j)) \tag{5.26}$$

wobei $\hat{m}$ das GMM des korrekten Sprechers referenziert. Bei anderer Betrachtungsweise entspricht die Abstandsberechnung in Gl. 5.25 einer Scorenormalisierung, wie sie bei Sprecherverifikationssystemen [Rey97, Nak97, Ros96] eingesetzt wird. Das Rechteckfenster, innerhalb dessen $LC_{GMM}(n)$ berechnet wird, entspricht dem Scoreakkumulationszeitraum für den die Verifikationsentscheidung getroffen wird.

MFCC12	MFCC24	MFCC36
-0.147	-0.065	-0.020

Tab. 5.6: Korrelationskoeffizient zwischen LC_{GMM} und LSR für GMM mit 64 NV bei unterschiedlicher Zusammensetzung des Merkmalsvektors.

Tabelle 5.6 zeigt eine, im Vergleich zu LC_{mean}, deutlich geringere Sprechgeschwindigkeitsabhängigkeit der sprecherweisen Konfidenz LC_{GMM}. Die Normierung auf ein sprecherunabhängiges Weltmodell führt also de facto zu einer Eliminierung der Sprechgeschwindigkeitsabhängigkeit bei der Modellierung der Mustervektoren. Die auftretende Abhängigkeit zwischen LC_{GMM} und LSR ist bei den rein statischen Merkmalen MFCC12 ausgeprägter als MFCC36. Ein Grund hierfür könnte in der starken SR-Abhängigkeit liegen, die bei Verwendung der MFCC36-Vorverarbeitung zwischen LAS und LSR zu beobachten ist. Bei MFCC36 wird die Auswirkung bei den rein statischen Merkmalen durch die ausgeprägtere bei den Delta-Koeffizienten überlagert und durch die Normalisierung nahezu vollständig eliminiert.

Bei Verifikationssystemen erfolgt die Entscheidung, ob es sich um den fraglichen Sprecher handelt, anhand des Vergleich des normalisierten Scores LC_{GMM} mit einem festzulegenden Schwellwert. Als Begründung für die Einführung der Normalisierung durch ein generisches SI-GMM wird die Relativierung des absoluten Scoreniveaus, das durch den nichtsprecherspezifischen (z.B. phonetischen) Inhalt der Testäußerungen beeinflusst ist, angegeben [Rey97, Col96, Iso99]. Obige Experimente zeigen deutlich, dass bei Systemen, die auf Merkmalsvektoren mit Deltakoeffizienten fußen, hierdurch insbesondere eine *Eliminierung der Sprechgeschwindigkeitabhängigkeit* mit einhergeht.

Der Einsatz dieses Konfidenzmaßes bietet sich demnach bei der Verifikation von Sprechern an, um den Sprechgeschwindigkeitseinfluss vernachlässigen zu können. Für den Einsatz als Distanzmaß zur Sprechergruppierung führt der Ansatz mit LC_{GMM} allerdings nur bedingt weiter, da beim Vergleich zweier Sprecher gilt:

$$
\begin{aligned}
LC_1 - LC_2 &= \frac{1}{N_F} \sum_{j=n-\frac{N_F}{2}}^{n+\frac{N_F}{2}-1} (S_1(\boldsymbol{x}_j) - S_{SI}(\boldsymbol{x}_j)) - \\
&\quad \frac{1}{N_F} \sum_{j=n-\frac{N_F}{2}}^{n+\frac{N_F}{2}-1} (S_2(\boldsymbol{x}_j) - S_{SI}(\boldsymbol{x}_j)) = \\
&= \frac{1}{N_F} \sum_{j=n-\frac{N_F}{2}}^{n+\frac{N_F}{2}-1} (S_1(\boldsymbol{x}_j) - S_2(\boldsymbol{x}_j)) \qquad (5.27) \\
LC_1 > LC_2 &\rightarrow \frac{1}{N_F} \sum_{j=n-\frac{N_F}{2}}^{n+\frac{N_F}{2}-1} S_1(\boldsymbol{x}_j) > \frac{1}{N_F} \sum_{j=n-\frac{N_F}{2}}^{n+\frac{N_F}{2}-1} S_2(\boldsymbol{x}_j)
\end{aligned}
$$

Die Distanzbestimmung bleibt also von der Normalisierung unberührt. Wenn allerdings das jeweilige Vergleichssprechermodell "2" zu Sprecher "1" das gleiche Verhalten zeigt, wie ein generisches SI-Modell, so wird dadurch ebenfalls eine Elimination der Sprechgeschwindigkeit erreicht. Verifiziert wird dies in Abb. 5.6. Es zeigt die Frame-weise Sprechererkennungsrate (IR) für die Xval96 in Abhängigkeit von der lokalen Sprechgeschwindigkeit (Modelle: 74 GMMs mit je 64 Basisfunktionen).

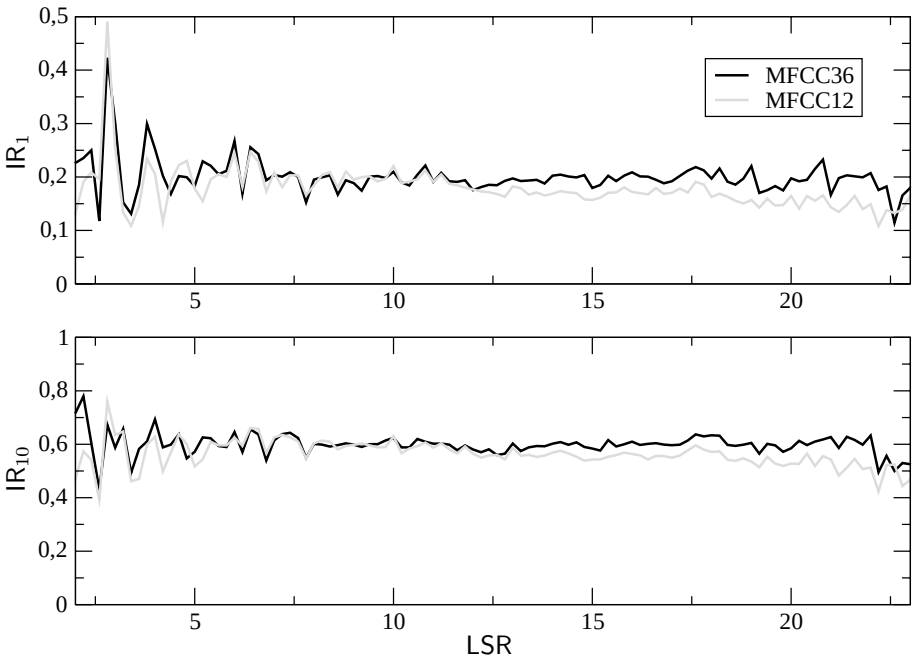

Abb. 5.6: Frameweise Identifikationsrate (IR_1) sowie Top-10 IR (IR_{10}) in Abhängigkeit von LSR und Struktur des Merkmalsvektors.

Die Analyse der beiden Verläufe in Abb. 5.6 legt ein unterschiedliches Verhalten offen. Bei der Betrachtung der Graphen der Modelle mit $\Delta\Delta$-Koeffizienten (MFCC36) zeigt sich

im Prinzip keine nennenswerte Restabhängigkeit von der Sprechgeschwindigkeit. Die Modelle halten ihre Erkennungsleistung über nahezu den gesamten LSR-Bereich hinweg. Lediglich bei einer LSR von über 20 Phoneme/s deutet sich ein leichter Abfall der Identifikationsrate an. Allerdings nimmt hier aufgrund der annähernden Gaussverteilung der Sprechgeschwindigkeit die absolute Häufigkeit des Auftretens solcher Merkmalsvektoren stark ab. Interessanterweise weisen die Modelle, die ausschließlich statische MFCC-Komponenten (keine Deltas) als Merkmale verwenden, einen deutlichen Abfall der Erkennungsleistung bei steigender LSR auf. Bis zu einer mittleren LSR von ca. 13 Phoneme/s haben diese Modelle noch in etwa die gleiche Performanz wie die Modelle mit Ableitungskoeffizienten, darüber jedoch beginnt die Performanz abzunehmen. Dies deckt sich mit den Ergebnissen in Tab. 5.6.

Als kritischer ist der absolute Wert der Kurven anzusehen. Der Wertebereich von IR_1 zeigt, dass bei den betrachteten 74 Sprechern nur ca. 20 % aller Merkmalsvektoren dem korrekten Sprecher zugeordnet werden können (vgl. auch Tab. 5.4) . Diese Aussage wird durch IR_{10} etwas relativiert, da der korrekte Sprecher immerhin zu ca. 60 % unter den 10 wahrscheinlichsten Hypothesen enthalten ist. Beide Graphen zeigen jedoch, dass *ein* einzelner Merkmalsvektor deutlich zu wenig ist, um eine robuste Aussage über Sprecherähnlichkeit zuzulassen.

5.3 Automatische Sprechergruppierung

Das Ziel der automatischen Sprechergruppierung ist, die Sprecher der Trainingsdatenbasis anhand gemeinsamer Eigenschaften oder Ähnlichkeitsbeziehungen zu Gruppen zusammenzufassen. Dieses sollte jedoch möglichst automatisch, ohne weitergehendes Wissen über die Sprecher erfolgen. Als Ansatz bietet sich hier die Verwendung iterativer Clusterverfahren - auf Basis definierter Abstandsmaße - an.

5.3.1 Sprecherabstand

Die in dieser Arbeit primär untersuchten Abstandsmaße zwischen 2 Sprechern i und j berücksichtigen die Modelle, sowie die verfügbaren Merkmalsvektoren der Sprecher. Im Falle der stochastischen GMM-Modellstrukturen lässt sich der asymmetrische Abstand D_{ij} definieren:

$$D_{ij} = -\frac{1}{N_P^i} \sum_{n=1}^{N_P^i} \log p_j(\boldsymbol{x}_n^i) \tag{5.28}$$

Alternativ lässt sich der Abstand beispielsweise auch über die relative Entropie (Kullback-Leibler Distanz) definieren [Foo94]. Die Mustervektoren eines Sprechers werden in Gl. 5.28 durch das Modell des zu vergleichenden Sprechers bewertet. Um 2 Abstände vergleichen zu können, muss eine Normierung auf die Anzahl N_P^i der in die Berechnung eingegangenen Vektoren erfolgen. In Gl. 5.28 erfolgt eine Invertierung der Summe, um den Term konsistent als Abstandsmaß behandeln zu können. Für VQ-basierte Modelle ergibt sich entsprechend:

$$D_{ij} = \frac{1}{N_P^i} \sum_{n=1}^{N_P^i} d_j(\boldsymbol{x}_n^i) \tag{5.29}$$

Das Abstandsmaß kann angenähert werden, indem statt der Mustervektoren eines Sprechers dessen Prototypmittelpunkte durch das Vergleichsmodell bewertet werden (Gl. 5.30).

$$D_{ij} = \frac{1}{K_i} \sum_{k=1}^{K_i} d_j(\boldsymbol{\mu}_{ik}) \quad \text{bzw.} \quad D_{ij} = -\frac{1}{K_i} \sum_{k=1}^{K_i} \log p_j(\boldsymbol{\mu}_{ik}) \tag{5.30}$$

Gl. 5.28 als auch Gl. 5.29 sind asymmetrisch, d.h. $D_{ij} \neq D_{ji}$ für $i \neq j$. Sie lassen sich jedoch durch Mittelung in symmetrische Abstandsmaße überführen:

$$D_{ij}^{symm} = \frac{1}{2}(D_{ij} + D_{ji}) \qquad \text{oder} \qquad D_{ij}^{symm} = \frac{N_P^i * D_{ij} + N_P^j * D_{ji}}{N_P^i + N_P^j} \tag{5.31}$$

die für den Fall $N_P^i = N_P^j$ identisch sind.

5.3.2 Gruppenmodell und -abstand

Ein gewichtiges Augenmerk liegt auf der Frage wie ein Sprechercluster, d.h. eine Gruppe ähnlicher Sprecher, repräsentiert wird. Diese Frage stellt sich sowohl für den Clustervorgang als auch für die Zuordnung in der Erkennungsphase (s. Abschnitt 5.2.2).

5.3.2.1 "Cluster"-Modell

Bei einem Cluster-Modell wird jede Gruppe durch ein eigenes, gemeinsam trainiertes VQ- oder GMM-Modell repräsentiert. Die Parameter dieses Modells werden nach einer Teilung, respektive Vereinigung, eines Clusters mit den Trainingsdaten der dem Cluster zugeordneten Sprecher neu nachgeschätzt. Dies kann für GMMs beispielsweise mittels des ML- oder MAP-Trainingsalgorithmus (s. Kap. 2) erfolgen. Der Abstand G_{rs} zweier Gruppen r und s ergibt sich analog zu Gl. 5.30.

5.3.2.2 Referenzsprecher

Beim Referenzsprecher-Ansatz (RS) wird eine Gruppe r durch einen definierten Sprecher repräsentiert. Als RS wird derjenige Sprecher ausgewählt, der von den übrigen Mitgliedern derselben Gruppe den geringsten Abstand gemäß des verwendeten Abstandsmaßes D_{ij} (s. Gl. 5.28 und 5.29) aufweist:

$$i_{RS}^r = \arg \min_{i^r=1..K_S^r} \sum_{j^r=1, j^r \neq i^r}^{K_S^r} D_{i^r j^r}^{\gamma} \tag{5.32}$$

Eine quadratische ($\gamma = 2$) Abstandsbewertung gewichtet im Vergleich zu einer linearen ($\gamma = 1$) große Abstände stärker, was zu einer Bevorzugung zentral gelegener Sprecher führt. Das Referenzsprecherkonzept kann in unterschiedlicher Ausprägung in die Cluster-zu-Cluster Abstandsberechnung einfließen. Im "Extremfall" wird der Gruppenabstand ausschließlich durch die Distanz der jeweiligen Referenzsprecher festgelegt.

$$G_{rs} = D_{i_{RS}^r j_{RS}^s} \tag{5.33}$$

Der eindeutige Vorteil dieses Ansatzes ist darin zu sehen, dass die einzelnen Abstände D_{ij} der Referenzen zueinander im voraus berechnet werden können. Zur Laufzeit des Clusteralgorithmus sind, im Gegensatz zur Cluster-Modell Repräsentation, keine Parameterneuschätzungen nötig. Die Laufzeit des reinen Clusteralgorithmus reduziert sich hierdurch von mehreren Tagen (PentiumII 400MHz) auf wenige Minuten. Das Referenzsprecherkonzept findet sich in ähnlicher Form beispielsweise bei Hazen [Haz00]. Er benutzt eine gewichtete Überlagerung aus RS-HMMs, um adaptierte Spracherkennungsmodelle für einen neuen Sprecher zu erzeugen. Die Gewichte werden, analog zur MLED-Schätzung, mit einer ML-Optimierung hergeleitet. Diese benötigt jedoch wiederum eine phonetische Segmentierung der Adaptionsdaten.

5.3.2.3 "Furthest-Neighbor"-Abstand

Die Verwendung von Minimum-Abstands-Klassifikatoren ist bei einer Clusterrepräsentation, in der die Streuung der Elemente eingeht (z.B. Cluster-GMM) ungünstig. Die Größe einer Gruppe beeinflußt in diesem Fall die Abstandsfestlegung und begünstigt die Tendenz ausgedehnter Cluster, im Verlauf einer Bottom-Up Gruppierung, weiter und v.a. stärker zu wachsen. Dies führt zu größenmäßig extrem unausgeglichenen Sprechergruppen. Ein Ausweg ist die Gewichtung des Abstandsmaßes durch die Clustergröße, so dass Abstände zwischen kleinen Gruppen bevorzugt werden. Bei statistischen Modellen entspricht dies im Prinzip dem sog. "Bayesian Information Criterion" (BIC) [ChS98, Cho99].

Ein anderer Weg ist die Verwendung einer Furthest-Neighbor-Abstandsdefinition. Der Abstand eines Sprechers zu einer Sprechergruppe wird nicht mehr durch den am nächsten gelegenen Sprecher dieser Gruppe, sondern durch den entferntesten definiert. Mit der Größe einer Gruppe wächst auch der Abstand zum entferntesten Element, was indirekt also die unerwünschte Bevorzugung großer Gruppen eliminiert. Der Abstand zweier Sprechergruppen r und s ergibt sich durch Überkreuzbewertung aller Sprecher K_S^r bzw. K_S^s der beiden Gruppen:

$$G_{rs} = \frac{1}{K_S^r} \sum_{i^r=1}^{K_S^r} \max_{j^s=1..K_S^s} D_{i^r j^s} \tag{5.34}$$

wobei unter Verwendung eines asymmetrischen Maßes D_{ij} auch hier gilt: $G_{rs} \neq G_{sr}$ für $r \neq s$. Dieses Abstandmaß kann auch beim Übergang auf das Referenzsprecherkonzept aufrecht erhalten werden:

$$G_{rs} = \max_{j^s=1..K_S^s} D_{i_{RS}^r j^s} \tag{5.35}$$

5.3.3 Verfahren zur Sprechergruppierung

Ziel der Sprechergruppierung ist das Auffinden von Sprechern in der Trainingspopulation mit ähnlichen Charakteristika im Merkmalsraum. Hintergedanke ist hierbei die Möglichkeit für jede Gruppe robuste, spezialisierte Erkenner-HMMs zu trainieren. Das Auffinden bzw. Zusammenfassen zu Gruppen sollte automatisch, ohne explizites Expertenwissen erfolgen. Für die automatische Gruppierung bietet sich die Verwendung iterativer Clusteralgorithmen an. Wie bereits erwähnt, sind aufgrund der vergleichsweise eingeschränkten Sprecherzahl

(ca. 1000) beide "Richtungen" der iterativen Gruppierung mit Hinblick auf die Rechenzeit möglich.

- Top-Down: fortgesetztes Teilen von Gruppen
- Bottom-Up: iteratives Zusammenfassen von Gruppen

In beiden Fällen werden die Gruppierungsoperationen (teilen bzw. zusammenfassen) solange fortgesetzt, bis entweder die gewünschte Zahl an Sprechergruppen erreicht ist, oder ein bestimmtes Abbruchkriterium erfüllt ist.

5.3.3.1 "Top-Down" Sprechergruppierung

Auf das grundlegende Prinzip der Top-Down Gruppierung wurde bereits im Vorfeld eingegangen (vgl. Kap. 2.2 und 4). Die Teilungsoperationen der Sprechergruppen erfolgen ebenso sequentiell und hierarchisch. Die Auswahl der Gruppe, die in einer Iteration zu teilen ist, kann anhand der Gruppenstärke oder der Intra-Gruppenstreuung festgelegt werden. Bei Verwendung des RS-Konzepts kann die Streuung als mittlere Abweichung bzgl. des RS (Gl. 5.32) ermittelt werden.

$$\sigma^{\gamma} = \frac{1}{K_S^r - 1} \sum_{j=1, j \neq i_{RS}^r}^{K_S^r} D_{i_{RS}^r j}^{\gamma} \tag{5.36}$$

5.3.3.2 "Bottom-Up" Sprechergruppierung

Bottom-Up Clusterverfahren sind gekennzeichnet durch ein fortgesetztes Zusammenfassen von Elementen, in diesem Fall Sprechergruppen. Den Ausgangspunkt bilden $K_G = K_S$ Cluster, wobei zu Beginn ein jeder nur mit einem einzelnen Sprecher besetzt ist. In jeder Iteration des Verfahrens werden die beiden ähnlichsten Gruppen zusammengefasst. Bei Verwendung eines Cluster-Modells zur Repräsentation einer Gruppe muss nach einer Vereinigungsoperation eine Neuschätzung des entstandenen Gruppenmodells erfolgen. Darüber hinaus müssen die betroffenen Gruppenabstände neu berechnet werden. Hier genügt es allerdings die Abstände des neuen Modells zu den verbleibenden Gruppen neu zu bestimmen. Da die Zahl der Sprechergruppen im Verlauf des Algorithmus linear abnimmt, sinkt der Aufwand für die Abstandsneuberechnung ebenfalls linear. Im Gegenzug steigt jedoch mit wachsender Gruppenstärke die Datenmenge innerhalb einer Gruppe, anhand derer die Parameter des Gruppenmodells geschätzt werden. Der Aufwand für das Neuschätzen entfällt bei Verwendung einer RS-basierten Abstandsdefinition. Hier müssen nach einer Vereinigungsoperation lediglich der neue RS bestimmt, sowie die betroffenen Gruppenabstände neu berechnet werden. Aufgrund der konstanten, vorberechneten Sprecherabstände ist der Aufwand hierfür vergleichsweise vernachlässigbar.

5.3.3.3 "K-Means" Sprechergruppierung

Eine Möglichkeit zur Erzeugung einer fixen Anzahl von Sprechergruppen bietet der sogenannte K-Means Algorithmus. In Kapitel 2.2 wurde dieses Verfahren zum Zwecke der Merkmalsvektorgruppierung vorgestellt. Er diente in diesem Zusammenhang zur Initialisierung der

HMM- bzw. GMM-Modelle. In ähnlicher Form kann der Algorithmus auch zur Sprechergruppierung eingesetzt werden. Der prinzipielle Ablauf des Algorithmus entspricht dabei dem in Kapitel 2.2 beschriebenen.

Zu Beginn werden K_G Sprecher willkürlich als Zentroid(=Referenz)sprecher ausgewählt. Dies entspricht dem initialen Ausgangszustand für den eigentlichen Algorithmus. Jeder der übrigen Sprecher wird dem nächstgelegenen Zentroidsprecher zugeordnet. Nun wird für jede der entstandenen Gruppen der Referenzsprecher gemäß Gl. 5.32 neu bestimmt. Im nächsten Schritt wird wieder jeder Nicht-Referenzsprecher dem nächstgelegenen Referenzsprecher zugeordnet. Diese Abfolge aus Neu-Zuordnen und Neubestimmung der RS wird solange wiederholt bis keine Änderung der Sprecher-zu-Referenzsprecher Zuordnung mehr auftritt. Experimentell führt die zufallsmäßige Initialisierung des Verfahrens jedoch zu sehr unausgeglichenen Sprechergruppen. Günstiger erweist sich eine Kombination mit den vorgenannten Gruppierungstechniken.

5.3.3.4 Kombinierte "K-Means" + "Bottom-Up/Top-Down" Sprechergruppierung

Die rein hierarchische Gruppierung anhand eines Bottom-Up bzw. Top-Down Ansatzes lässt nur eine starre Sprecher-zu-Sprechergruppe Zuordnung zu, d.h. jeder Sprecher der einmal einer Gruppe zugeordnet wurde, verbleibt in dieser Gruppe bzw. den davon abgeleiteten Gruppen. Die Zugehörigkeit wird nur durch die Vereinigung bzw. Teilung einer Gruppe verändert. Durch die Kombination mit dem K-Means Ansatz kann die fixe Zuordnung "aufgeweicht" werden, indem nach jeder Teilungs-/Vereinigungsiteration eine K-Means Optimierung durchgeführt wird. Durch die zusätzliche K-Means Neuordnung werden speziell ungünstig liegende Sprecher einer besser geeigneten Gruppe zugeschlagen. Dies bestätigt sich auch experimentell, da bereits nach maximal 2 Iterationen des K-Means keine Änderung mehr zu beobachten ist.

5.3.4 Vergleich mit fixer Gruppeneinteilung anhand des Vokaltrakts

Die Modellierung und Gruppierung von Sprechern durch ihre Repräsentation im Merkmalsraum ist zwar im Hinblick auf die Erkennung günstig, da im Prinzip die gleichen Merkmale (s. Abb. 1.4) zugrundeliegen. Eine eingängige, anschauliche Interpretation der resultierenden Gruppen wird hierdurch jedoch erschwert. Eine Alternative ist die Charakterisierung von Sprechern anhand realer physikalischer Eigenschaften des Vokaltrakts [Nai98].

Ein Sprecher kann durch die Länge seines Vokaltrakts, die direkte Auswirkungen auf die sprachlichen Charakteristika des Menschen hat, beschrieben werden. Diese Form der Modellierung wurde auch zur Sprecheradaption in Form der sogenannten Vokaltraktlängennormierung (VTLN, engl. 'vocal tract length normalization') vorgeschlagen [LeL96, Pfa00b]. Bei der Normierung wird der umgekehrte Weg beschritten: statt individuelle Modelle zu trainieren, wird ein Sprecher anhand einer Transformationsvorschrift auf einen Normsprecher abgebildet. Im Rahmen der VTLN wird im Frequenzspektrum eine Verzerrungsfunktion berechnet ('frequency warping'), anhand derer das, durch die unterschiedliche Vokaltraktlänge verzerrte,

sprecherspezifische Spektrum auf ein Normspektrum zurückgeführt werden kann. Diese Verzerrungsfunktion ist charakterisiert durch den Parameter α, der für jeden Sprecher individuell zu bestimmen ist. Um ein System an einen neuen, unbekannten Anwender anzupassen, muss dessen Verzerrungs(=Warping)faktor ermittelt werden. Im Rahmen dieser Arbeit konnte, wie auch von anderen Autoren [Six00], gezeigt werden, dass hierzu ein GMM-basiertes Klassifikationssystem (s. Abb. 5.3) geeignet ist.

Die Idee ist den Wertebereich des Warpingfaktors α zu quantisieren. Der Bereich $\alpha = 0.88 \ldots 1.12$ wird zumeist in 12 Segmente unterteilt, d.h. $\Delta\alpha = 0.2$. Für jeden Sprecher der Trainingsdatenbank wird mittels einer ML-Schätzung der für ihn gültige Referenzparameter α bestimmt [Pfa00b]. Anhand dieses Wertes kann jeder Sprecher eindeutig einem Segment zugeordnet werden. Für jedes der Segmente wird mit den, auf diese Weise indirekt zugeordneten, Trainingsdaten ein eigenes GMM-Modell ($\hat{=}$Cluster-GMM) trainiert. Mittels der so gewonnenen Modelle kann für einen Testsprecher eine 'Segment'- Klassifikation erfolgen - was gleichbedeutend ist mit der Bestimmung des Warpingfaktors.

In Versuchen konnte gezeigt werden, dass der auf diese Art und Weise ermittelte Warpingfaktor mit dem mittels ML-Schätzung bestimmten Referenzwert eine Korrelation von $\rho \approx 0.92$ aufweist. Die Erkennungsrate zeigte nur geringfügige Verschlechterung im Vergleich zu der bei ML-Bestimmung.

Unabhängig von der Verwendung für den VTLN-Ansatz stellt die Einteilung des Warpingfaktors in Segmente unmittelbar eine Gruppeneinteilung der Sprecher dar. Für jede dieser Gruppen kann mit den zugehörigen Sprachdaten somit auch ein eigener HMM-Modellsatz erzeugt werden. In der Erkennungsphase lässt sich mittels des GMM-basierten α-Klassifikationssystems auch eine Entscheidung über die zu verwendenden Gruppenmodelle treffen.

5.3.4.1 Experimente

Abbildung 5.7 zeigt einen typischen Verlauf der WER auf dem Eval96-Testset in Abhängigkeit von der Sprechergruppenzahl. Als Basis der Sprechergruppierung wurden GMM-Modelle mit 64 Prototypen vorausgesetzt. Der Gruppierungsvorgang erfolgte Bottom-Up mit zusätzlicher K-Means Neuordnung. Für jede der entstandenen Sprechergruppen wurden individuelle HMM-Modelle trainiert. Die Zuordnung während der Erkennung erfolgte satzweise, wahlweise implizit anhand des besten HMM-Scores oder explizit, anhand des ähnlichsten RS-Modells. In Abb. 5.7 erscheint zusätzlich der Verlauf der absolut besten WER, der sich durch Auswahl der Sprechergruppe ergibt, die je Satz den niedrigsten Fehler erzielt.

Die Berücksichtigung von Sprechergruppen erlaubt eine deutliche Verringerung der Wortfehlerrate von bis zu 6.7% relativ. Mit zunehmender Gruppenzahl steigt allerdings die Schwierigkeit eine geeignete Sprechergruppe auszuwählen. Insbesondere die explizite Selektion führt hier zu einem deutlichen Anstieg der WER. Eine frühzeitigere explizite Modell-Selektion bereits nach $1s$ weist gegenüber der gezeigten Selektion (nach Turnende) eine geringfügig schlechtere WER auf. Dieses Ergebnis ist jedoch nicht als kritisch anzusehen, da die Entscheidung für eine Gruppe nicht zwingend hart getroffen werden muss. So kann beispielsweise nach

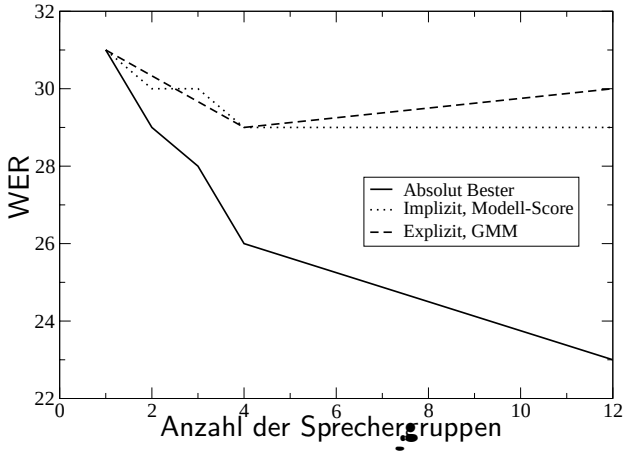

Abb. 5.7: Verlauf der WER (in [%]) bei unterschiedlicher Clusteranzahl und -selektion nach bzw. vor der Erkennung.

1s die Anzahl der bewerteten Sprechergruppen reduziert werden, indem unwahrscheinliche Gruppen geprunt werden. Mit zunehmender Länge der Bewertungssequenz [He99a] kann die Zahl der bewerteten Gruppen weiter eingeschränkt werden.

Auffallend in Abb. 5.7 ist der Verlauf der WER bei *optimaler* Selektion anhand des minimalen Fehlers. Er vermittelt einen Eindruck über das - zumindest theoretisch - erreichbare Fehlerniveau. Analysiert man bei 12 Sprechergruppen die Hypothesen-Liste der expliziten GMM-Auswahl, so zeigt sich, dass die jeweilige Worthypothese mit dem minimalen Fehler zu mehr als 70% von den wahrscheinlichsten 3 Sprechergruppen erzeugt wird. Versuche mit einer ROVER-ähnlichen, konfidenz-basierten [Fis97, Lue00] Kombination der Einzelhypothesen ließen allerdings nur eine geringfügige weitere Verringerung der WER zu.

Vergleicht man hierzu die Einteilung anhand der VTLN-Klassen (s. Abschnitt 5.3.4), so zeigt diese eine etwas schlechtere Gesamtperformanz (12 Gruppen, WER: 29.7% bei impliziter, und 30.0% bei expliziter Selektion) als die automatische Gruppierung. Die "optimale" WER erreicht bei 12 Gruppen jedoch ebenfalls einen Wert von 23.8% absolut.

5.3.5 "Subspace Clustering"

Eine robuste Möglichkeit zur Sprechergruppierung liegt in der Reduktion auf einen Subraum des originalen Merkmals- oder Modellraums [Fal01b]. Bei den bisherigen Betrachtungen wurde die Sprecheridentifikationsleistung von Modellen als Indikator für die "Qualität" von Modellstrukturen verwendet. Häufig wird jedoch die exakte Identifikation eines Einzelsprechers durch ungünstig geschätzte Parameter negativ beeinflusst. Wünschenswert wäre daher eine Ähnlichkeitsbestimmung zwischen den Sprechern, die nur durch primäre Modellunterschiede getragen wird. Durch eine geeignete Transformation kann die im Unterraum verbleibende Information auf die wesentliche, die Sprecher charakterisierende, Information reduziert werden. Solange durch die Transformation bzw. Reduktion die Ähnlichkeitsbeziehungen erhalten bleiben, kann die Gruppierung auch in diesem Unterraum stattfinden.

Als Prototyp einer solchen Transformation wurde hier die sogenannte Eigenvoice Methode untersucht (s. Abschnitt 2.5). In [Thy00] wurde dieser Ansatz für ein reines Sprechererkennungsexperiment herangezogen. Die Autoren trainierten mittels MLED sprecherspezifische GMMs, die bei sehr wenig Trainingsdaten konventionellen, ML-trainierten GMM leicht überlegen sind. In [Fal01a] konnte darüber hinaus anhand des in Abschnitt 5.2.1.4 beschriebenen Systemaufbaus gezeigt werden, dass durch die Einführung individueller Eigenraumkoeffizienten für verschiedene, phonetisch motivierte Teile des "Supervektors" bessere Performanz bzgl. der Sprecheridentifikationsrate erzielt werden kann als mit globalen Koeffizienten.

Die Eigenvoice Transformation basiert auf einer Auswertung der Varianz der Sprechermodellparameter im Merkmalsraum. Wie in Abschnitt 2.5 ausgeführt, werden zu diesem Zweck für jeden Sprecher eigene Modelle trainiert. Die Transformation ist im Prinzip mit GMMs [Thy00], HMMs [Kuh98, Kuh00, Ngu99], oder Abwandlungen beider Modellstrukturen [Fal01a] möglich. Für das Training der individuellen Sprechermodelle eignet sich der MAP-Algorithmus, da er durch den Bezug auf das gemeinsame SI-Ausgangsmodell die korrekte Zuordnung der Modellparameter gewährleistet. In Abwandlung hiervon wurde aber auch der MLLR-Algorithmus, direkt [ChK00], oder in Kombination mit MAP [Bot00], zum Training vorgeschlagen. Allerdings ist die theoretische Rechtfertigung für den Einsatz einer MLLR eher fraglich, da direkte Relation der Modellparameter (s. Abb. 2.1 und 2.2) durch die Transformation nicht mehr unmittelbar gegeben ist. [ChK00] nimmt hierzu an, bzw. setzt voraus, dass die Koeffizienten der MLLR-Transformationsmatrizen die sprecherspezifische Information tragen. In [Wan01] gehen die Autoren sogar so weit, diese Koeffizienten als Merkmale für die Sprecheridentifikation zu verwenden.

Kern der Eigenvoice Transformation ist die Hauptachsenanalyse der Sprechermodellstreumatrix $\boldsymbol{C}_Z$ (Gl. 2.42). Da der Rang von $\boldsymbol{C}_Z$ durch die Anzahl der Sprecher K_S beschränkt wird, ergeben sich nur $Rg(\boldsymbol{C}_Z) = K_S - 1$ linear unabhängige Eigenvektoren nach der PCA-Analyse. Werden keine weiteren Eigenachsen weggelassen, so ist die Dimension des reduzierten Raums auf $K_S - 1$ begrenzt.

Ein kritischer Aspekt bei der Bildung von Sprechergruppen ist darin zu sehen, geeignete Modelle und Abstandsmaße zu finden bzw. zu definieren. Vorgestellt wurden in dieser Arbeit speziell GMM- und HMM-basierte Ansätze. Selbst bei der kompakten Mixturverteilung, wie sie beim GMM angenommen wird, muss bereits ein "Umweg" über eine Likelihood-Abstandsdefinition gemacht werden. Einfache Abstandsmaße zwischen Sprechern in der Form $d_{ij} = |\boldsymbol{\mu}_i - \boldsymbol{\mu}_j|$ lassen sich nicht unmittelbar angeben. Bei strukturell komplexeren Modellen, wie beispielsweise einem HMM, kommen noch weitere Probleme hinzu. Durch die Unterteilung des Modells in Einzelzustände wird die Abstandsmessung zwischen zwei Modellen noch um die Zuordnung der jeweiligen Abschnitte erschwert [Gao97]. Eine 1:1 Zuordnung der Zustände berücksichtigt nicht, dass nach der Parameterschätzung bei verschiedenen Sprechern u.U. unterschiedliche Zeitabschnitte in den Zuständen repräsentiert werden. Innerhalb der Einzelzustände werden wiederum Mixturverteilungen eingesetzt. Für die Festlegung von Abstandsmaßen zwischen Sprechern müssen hier wiederum Likelihood Betrachtungen [Jua85],

oder nichtlineare Abstandsmaße wie beispielsweise $d_{ij} = \min_k |\boldsymbol{\mu}_{ik} - \boldsymbol{\mu}_{jk}|$ herangezogen werden. Dieses stückweise lineare Abstandsmaß kann bei VQ-Modellen oder Mixturverteilungen, bei denen nur die Mittelpunkte betrachtet werden, angewandt werden. Nachteilig ist jedoch, dass bereits hier eine Suchoperation integriert ist. Wünschenswert wäre eine *robuste* Abstandsberechnung in der Form $d_{ij} = |\boldsymbol{\mu}_i - \boldsymbol{\mu}_j|$.

Ermöglicht wird dies durch die Anwendung der Eigenvoice-Transformation (s. Abschnitt 2.5). Ziel dieser Transformation ist die Varianz in den Sprechermodellen zu erfassen und diese als zusätzliche Wissensquelle für die Parameterrestriktion im Rahmen einer Adaption auszunutzen (s. Kap. 2.5.3). Hierzu werden die Hauptachsen der Sprecherkovarianzmatrix $\boldsymbol{C}_Z$ mittels einer PCA (oder ICA, LDA) bestimmt. Die Matrix $\boldsymbol{C}_Z$ wird aus den Sprecher-Supervektoren gebildet. Ein Supervektor ergibt sich durch die sequentielle Ausrichtung und Zusammenfassung *aller* Modellparameter in einem einzigen Vektor, wobei jedoch meist nur die Mittelpunktsparameter herangezogen werden. Aus diesen Vektoren lässt sich die Sprecherkovarianzmatrix anhand Gl. 2.42 berechnen. Die Kovarianzmatrix erfasst lineare Abhängigkeiten zwischen den Einzelkomponenten des Supervektors, d.h. der Modellparameter. Die Anwendung der PCA führt zu einem neuen Koordinatensystem, in dem die Merkmale dekorreliert sind. Als Ergebnis erhält man Basisvektoren, die den Hauptstreurichtungen der Parameter entsprechen. Die Reihenfolge der Achsen nach der Transformation ist durch die Varianz in dieser Richtung gegeben, d.h. es lässt sich festlegen welcher Fehler durch das Weglassen von Achsen [Rus94] entsteht (s. Abb. 2.5).

Der Vorteil der Eigenvoice Transformation kann darin gesehen werden, dass die Modellparameter in einen niedrigdimensionalen Subraum transformiert werden, der primär durch die Varianz der Parameter gekennzeichnet ist. Etwaige Ausreißer durch nicht robust genug geschätzte Unbekannte, die sich möglicherweise nachteilig auf die Abstandsbestimmung auswirken, können durch die Restriktion auf den Eigenraum wirkungsvoll kompensiert werden. Nach der Transformation in den Raum der durch die Eigenvoices aufgespannt wird, reduziert sich ein Sprechermodell auf einen Eigenraumkoordinatenvektor $\boldsymbol{w}^s = [w_1^s, w_2^s, \ldots, w_{K_{EV}}^s]^T$ je Zustand. Die ursprünglichen Modelle können aus dem reduzierten Koordinatenvektor rekonstruiert werden durch:

$$\boldsymbol{\mu}_{sk} = \boldsymbol{m}_0^{sk} + \sum_{n=1}^{K_{EV}} w_n^s \boldsymbol{e}_n^{sk} = \boldsymbol{m}_0^{sk} + \boldsymbol{E}^{sk}\boldsymbol{w}^{sk} \tag{5.37}$$

mit $\boldsymbol{E}^{sk} = [\boldsymbol{e}_1^{sk}\ \boldsymbol{e}_2^{sk}\ \ldots \boldsymbol{e}_{K_{EV}}^{sk}]$. $\boldsymbol{e}_n^{sk}$ ist der Teilvektor des n-ten (Super)Eigenvektors $\boldsymbol{e}_n^s$ der zu Zustand s (im Falle von GMMs: $N_S = 1$) und Mittelpunkt k gehört. Eine vollständige Rekonstruktion des Originalmodells ist allerdings nur möglich, wenn $K_{EV} = Rang(\boldsymbol{C}_Z)$. In diesem Fall wäre die Abstandsberechnung nach Gl. 5.38 (numerische Aspekte unberücksichtigt) identisch mit der Berechnung im Originalraum. Ist K_{EV} kleiner als der Rang der Sprecherkovarianzmatrix, erfolgt eine Projektion auf die verbleibenden Eigenvoice Achsen. Die Information, die in den verworfenen Achsen enthalten ist, geht verloren. Durch die Transformation und Reduktion werden die Sprecherrepräsentationen auf die Raumrichtungen beschränkt, welche die relevanteste Sprecherinformation tragen. Dadurch wird eine direkte Modell-zu-Modell Abstandsberechnung möglich, die robust genug ist.

Im folgenden wird von GMM-Modellen und einem einzigen Koordinatenvektor $\boldsymbol{w}$ ausgegangen.

$$d_{ij}^2 = |\boldsymbol{w}_i - \boldsymbol{w}_j|^2 = (\boldsymbol{w}_i - \boldsymbol{w}_j)^T(\boldsymbol{w}_i - \boldsymbol{w}_j) = \sum_{n=1}^{K_{EV}} (w_{in} - w_{jn})^2 \tag{5.38}$$

Der Abstand d_{ij} zwischen zwei Sprechern i und j kann als Euklidischer Abstand im Eigenraum angegeben werden. Für die automatische Sprechergruppierung müssen aus dieser Sprecher-zu-Sprecher Abstandsfestlegung Gruppenabstandsmaße abgeleitet werden. Hierzu kommen aus Abschnitt 5.3.2 insbesondere die Verwendung eines Referenzsprechers, sowie das Furthest-Neighbor-Kriterium in Frage. Das Training eines eigenen Cluster-GMMs ist aufgrund der fehlenden Trainingsmenge nicht möglich. Im Gegensatz zum Originalraum, wo für jeden Sprecher die zugehörigen Trainingsmustervektoren vorliegen, ist im reduzierten Eigenraum nur der einzelne Sprecherkoordinatenvektor $\boldsymbol{w}$ als "Mustervektor" verfügbar. Für die automatische Gruppierung können ohne Änderung die Clusterverfahren aus Abschnitt 5.3.3 eingesetzt werden.

Darüber hinaus erlaubt die Eigenvoice-Transformation ein weiteres naheliegendes Prinzip der Sprecherzusammenfassung.

Binäre Sprechergruppierung:

Nach der Eigenvoice Transformation, d.h. im Eigenraum, sind die Sprechermodelle mittelwertsbefreit und entlang der Achsen verteilt (dekorreliert). Dies erlaubt eine "binäre", hierarchische Strategie zur Gruppenbildung. Jede Achse kann dabei als Entscheidungsfunktion aufgefasst werden:

$$c_k = \begin{cases} 1 & \text{wenn } w_k \geq 0 \\ 0 & \text{sonst} \end{cases} \tag{5.39}$$

Bei K_{EV} Eigenvoices ergibt sich für jede Achse eine Entscheidungsfunktion, insgesamt also ebenfalls K_{EV}. Mit jeder Entscheidungsfunktion ist genau eine Trennfunktion zwischen zwei Klassen - hier Sprechergruppen - verbunden. Bei K_{EV} Trennfunktionen können demnach $2^{K_{EV}}$ Klassen unterschieden werden. Eine Klassengrenze ergibt sich bei $w_k = 0$, also bei einem Vorzeichenwechsel eines Eigenvoice-Koeffizienten. Die Eigenachsen sind nach fallender Varianz sortiert, d.h die Sprecherverteilung entlang der ersten Achse weist die größte Streuung auf. Die Varianz sinkt mit jeder Achse, die zur Gruppenunterscheidung aufgenommen wird. Eine derartige Einteilung entspricht einem binären Entscheidungsbaum.

5.3.5.1 Experimente

Die Subspace-Gruppierungsexperimente wurden mit den 613 Sprechern des Verbmobil Korpus durchgeführt. Für jeden der Sprecher wurde ein eigenes GMM mit 64 NV mittels MAP aus einem generischen SI-Basismodell abgeleitet. Als Vorverarbeitung wurde MFCC24 zugrundegelegt. Ausgehend von den 613 Sprechern wurde der Eigenraum der Streumatrix $\boldsymbol{C}_Z$ berechnet. Die Dimension des Supervektors beträgt $64 * 24 = 1536$. Im Gegensatz zur Eigenvoice-Analyse eines HMM-basierten Spracherkennungssystems (vgl. Tab. 2.5) bleibt der

Berechnungsaufwand zur Eigenwertzerlegung der 1536×1536 Matrix handhabbar. Reduziert man den Eigenraum auf die ersten beiden $K = 2$ Eigenvoices, so ergibt sich die charakteristische Verteilung in Abb. 5.8. Für die Erkennungsexperimente wurden mit MAP individuelle HMM-Sätze für jede Gruppe erzeugt.

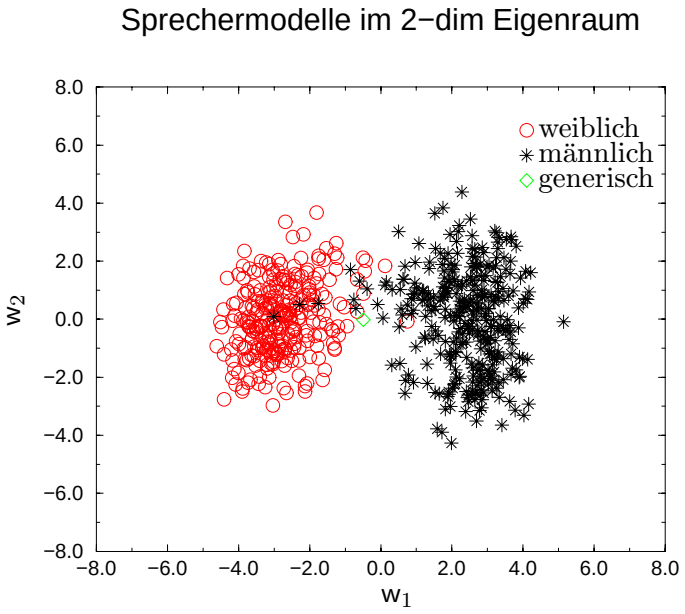

Abb. 5.8: Eigenraum mit $K_{EV} = 2$ Eigenvoices: starke Trennung in männliche ($\star$) und weibliche ($\circ$) Sprecher.

Der Anschaulichkeit halber wurde bei der Darstellung der Sprecherkoordinaten das Geschlecht des Sprechers berücksichtigt. Abb. 5.8 kann bereits optisch eine starke Trennung zwischen weiblichen und männlichen Sprechern entnommen werden. Die Raute kennzeichnet die Position des generischen, sprecherunabhängigen Basismodells. Die Position des SI-Basismodells unterscheidet sich geringfügig vom Ursprung, da dieser dem Mittelwert der adaptierten SD-Modelle entspricht. Der Eigenraum umfasst hier die ersten beiden Eigenvoices ($K_{EV} = 2$). Die erste Eigenvoice erklärt primär die geschlechtsspezifische Varianz in den Sprechermodellen.

Abb. 5.8 zeigt 3 männliche Sprecher, die tief im Bereich des anderen Geschlechts liegen. Bei der Symboldarstellung wurde (bewusst) die zum Zeitpunkt des Modelltrainings vorliegende Sprecherliste des Verbmobil Korpus, die auch das Sprechergeschlecht einschließt, zugrundegelegt. Eine Nachbetrachtung der betreffenden Sprecher zeigte, dass es sich offensichtlich um Frauen handelt, d.h. in der Trainingsliste also Verschriftungsfehler vorlagen. Das Eigenvoice-Verfahren ermöglicht bezüglich des Sprechergeschlechts eine schnelle, semiautomatische Überprüfung der Korrektheit dieser Zusatzinformation. Bei großen, nicht-selbst verschrifteten Sprachkorpora ist die Überprüfung der Korrektheit i.d.R. nicht mit vertretbarem Aufwand möglich. Aber gerade die Kenntnis des Sprechergeschlechts ist für das Training geschlechtsabhängiger Erkennermodelle grundlegend.

Ein großer Vorteil der binären Sprechergruppierung kann darin gesehen werden, dass die Einteilung unmittelbar durch die reduzierten Koordinaten gegeben ist und damit keinerlei Stabilitätsprobleme auftreten. Bei iterativen Clusteransätzen besteht bei ungünstiger Wahl der Abstandsmaße u.U. die Gefahr der Divergenz des Algorithmus. Dies äußert sich meist in einer sehr ungleichen Verteilung der Gruppenstärke. Aufgrund der fixen Einteilung kann ein derartiges Abweichen bei binärer Einteilung nicht auftreten. Durch die Trennung bei $w_i = 0$ wird eine gleichmäßige Gruppenstärke erreicht. Der Nachteil des binären Ansatzes zeigt sich in Abb. 5.8. Die Randverteilung bei Projektion auf die erste Eigenachse ergibt zwei annähernde Gaussverteilungen, die durch einen Schnitt bei $w_1 = 0$ sehr gut getrennt werden können. Die Randverteilungen bei Projektion auf die höheren Eigenachsen lassen sich jedoch durch eine unimodale Gaussverteilung mit Mittelpunkt bei $w_i = 0$ (für $i > 1$) beschreiben. Die Binärteilung trennt die Gruppen daher direkt im Maximum der Verteilung, was dazu führt, dass sehr viele ähnliche Sprecher auf unterschiedliche Gruppen aufgeteilt werden.

Gruppierungstechnik	K_{EV}	K_G	WER [%]
Bottom-Up	2	2	29.7
	2	4	29.2
	2	8	29.1
Binär	2	4	30.7
SI (Basissystem)	-	(1)	31.2
GMM-orig	-	12	29.2

Tab. 5.7: Vergleich der WER bei verschiedenen Gruppierungstechniken und Zahl der Sprechergruppen.

Die Ergebnisse in Tab. 5.7 bestätigen, dass durch die Ausbildung von 4..12 charakteristischen Sprechergruppen eine deutliche Reduktion der Wortfehlerrate von bis zu 2.1% absolut erzielt werden kann. Die binäre Einteilung erreicht aus den genannten Gründen bei gleicher Gruppenzahl nicht die Performanz der automatischen Einteilung, aber im Vergleich zum Basissystem ist dennoch eine Reduktion der WER möglich. Die Gruppierung im Subraum führt zu einer äquivalenten Performanz wie die Gruppierung im Originalraum (vgl. Abb. 5.7), bei deutlich robusterer Gruppenbildung. Für die Gruppenselektion in der Erkennungsphase wurde der in Abb. 5.3 dargestellte Systemaufbau eingesetzt, wobei ein GMM je Gruppe im Originalraum erzeugt wurde. Prinzipiell lässt sich der Gruppierungsvorgang auch auf Basis von HMM-Modellen durchführen. In diesem Fall können die Erkennermodelle - nach Mittelung der Subraumkoeffizienten - direkt durch Rückprojektion (Gl. 5.37) in den Originalraum erzeugt werden - ein nachfolgendes Training der HMM-Modelle entfällt.

Zusammenfassend lässt sich sagen, dass die Ausbildung von Sprechergruppen eine schnelle und v.a. robuste Anpassung an den aktuellen Sprecher ermöglicht. Ausgehend von dieser verbesserten Ausgangsbasis kann durch weitergehende Adaptionstechniken eine verfeinerte Angleichung erreicht werden.

Kapitel 6

Diskussion und Ausblick

Im Rahmen dieser Arbeit wurde die Anpassung eines automatischen Spracherkennungssystems an veränderte, sprecherspezifische Rahmenbedingungen näher betrachtet. Die Untersuchungen konzentrieren sich auf die Veränderungen der Spracheingabemuster, die durch unterschiedliche Sprecher, sowie deren variable Sprechgeschwindigkeit verursacht werden. Als Kernkonzept, um ein System robust gegen solche Abweichungen zu gestalten, wurde das Prinzip der "(Modell)Gruppenbildung" verfolgt. Die Gruppenbildung wurde hierbei auf verschiedenen Ebenen der stochastischen Modellierung untersucht.

Bei ASR-Systemen werden i.d.R. Hidden-Markov-Modelle (HMM) zur Repräsentation einzelner Spracheinheiten (hier Phoneme) verwendet. Innerhalb der Zustände eines HMMs wird mittels Gauss'scher Normalverteilungen die Verteilung der charakteristischen Mustervektoren der Spracheinheit im Merkmalsraum parametrisch nachgebildet. Bei sprecherunabhängigen Systemen werden die Parameter der Lautmodelle, d.h. insbesondere die Parameter der Gaussverteilungen, aus einer Sprachstichprobe von möglichst vielen Sprechern geschätzt.

Bei der Generierung der Lautmodelle greift das Konzept der "Gruppierung" auf verschiedenen Ebenen der Modellbildung. Auf der untersten Ebene betrifft dies die Mustervektoren, die innerhalb eines HMM-Zustands zusammengefasst - "gruppiert" - und durch Normalverteilungsprototypen repräsentiert werden. Auf dieser Ebene steht allerdings weniger die Gruppierung als solche im Vordergrund, sondern eher die Festlegung der Gruppenzahl (=Anzahl der Prototypen). Von der nächsthöheren Ebene der Gruppierung sind komplette Lautmodelle bzw. Teile von diesen, d.h. Zustände betroffen. Auswirkungen hat dies v.a. bei kontextabhängiger Modellierung: für jede Lauteinheit werden mehrere Modelle geschätzt - abhängig von den vorhergehenden und nachfolgenden Phonemen. Bei ausschließlicher Betrachtung des unmittelbaren rechten und linken Kontexts ergeben sich theoretisch N_{Ph}^3 verschiedene Modelle. Da wiederum nicht alle dieser Lauttripel in den Trainingsdaten (nur ca. 10 Prozent) gesehen werden, müssen die Modelle zu Gruppen zusammengefasst werden, um die robuste Schätzung der Systemparameter sicherzustellen. Dies geschieht vorzugsweise mit Entscheidungsbaumverfahren. Das Hauptaugenmerk dieser Arbeit ist auf die Erstellung robuster Erkennermodelle gerichtet - speziell mit Hinblick auf variable Sprechgeschwindigkeit. In diesem Zusammenhang wurden die Zustandsgruppierungstechniken daraufhin untersucht,

wie die Information über die vorliegende Sprechgeschwindigkeit gewinnbringend eingearbeitet werden kann. Desweiteren wurden die verschiedenen Möglichkeiten einer expliziten Klassen(=Gruppen)einteilung systematisch untersucht und verglichen.

Um die wertemäßig kontinuierliche Sprechgeschwindigkeit in Klassen fassen zu können, wurde der Wertebereich in definierte Bereiche eingeteilt. Mit dieser Einteilung geht die Trennung der Trainingsdaten einher. Für jede dieser Klassen lassen sich individuelle Modelle trainieren. Bei der Generierung robuster, kontextabhängiger HMM-Modelle stellt sich das Problem der Zustandsgruppierung. Hierbei wurde insbesondere das Entscheidungsbaum-Prinzip näher betrachtet. Bei diesem Verfahren ergeben sich unterschiedliche Ansatzpunkte, um eine Klasseneinteilung vorzunehmen. So kann auf den ungeteilten Trainingsdaten ein gemeinsamer Entscheidungsbaum erstellt werden, dessen Verteilungen dann separat trainiert werden. Für das Training der Verteilungen ergeben sich in diesem Fall zwei "Varianten". Die Verteilungen können unmittelbar mit den klassenspezifischen Daten geschätzt werden oder erst nach einigen Trainingsiterationen mit den gesamten Daten. Letzteres entspricht einem Nachtraining von sprechgeschwindigkeitsunabhängigen Modellen.

Wird die Datentrennung schon während des Entscheidungsbaumverfahrens berücksichtigt, so kann eine eigene Zustandsgruppierung für jede Sprechgeschwindigkeitskategorie ermittelt werden. Diese müssen jedoch dann unmittelbar mit den klassenweisen Daten trainiert werden. Vergleicht man die beiden Ansätze anhand der erzielbaren Worterkennungsraten, so zeigt sich deutlich, dass eine frühzeitige Trennung der Daten zu deutlichen Verschlechterungen führt. Das gleiche gilt, wenn für das kategorieweise Training ein direktes Maximum-Likelihood Verfahren zur Parameterschätzung eingesetzt wird. Eine Zwischenstufe wird erreicht, wenn zwar ein gemeinsamer Basisentscheidungsbaum auf allen Daten generiert wird, dieser jedoch mittels der individuellen Daten für die jeweilige Klasse zurechtgeschnitten wird. Dieses Vorgehen bietet den Vorteil, dass, basierend auf dem Basisbaum, vollwertige Erkennermodelle erstellt werden können. Diese wiederum sind den rudimentären Modellen, die zur Erzeugung des Entscheidungsbaums verwendet werden, an Genauigkeit überlegen und können eignen sich daher sehr gut zur Baumbeschneidung.

Eines der grundlegenden Probleme bei der Einteilung in Sprechgeschwindigkeitskategorien kann in der annähernden Gaussverteilung der Sprechgeschwindigkeit gesehen werden. Bei der angegebenen Einteilung der Klassen entfällt nur ein geringer Anteil der Trainingsdaten auf die Randkategorien "schnell" und "langsam". Dies erschwert die robuste Parameterschätzung. Darüber hinaus übt die Variation der Sprechgeschwindigkeit nicht auf alle Spracheinheiten den gleichen Einfluss aus. Einige werden sehr stark beeinflusst, andere nahezu gar nicht. Speziell für letzteren Fall macht die fixe Klasseneinteilung wenig Sinn, da sie nur zu einer Teilung der Daten führt, die die robuste Parameterschätzung erschwert. Um dem Rechnung zu tragen, wurde das Konzept des phonetischen Entscheidungsbaums um generelle Entscheidungen - wie beispielsweise Sprechgeschwindigkeit - erweitert. Das Verfahren kann daher selbständig anhand der gegebenen Datenverteilung entscheiden, ob eine Teilung dergestalt sinnvoll ist. Experimente mit diesem Ansatz zeigen, dass die, die Sprechgeschwindigkeit betreffenden, Entscheidungen zwar nicht die Bedeutung der des Geschlechts aufweisen, jedoch sehr wohl

auftreten. Allerdings ist die Bedeutung maßgeblich durch die Festlegung der Klassengrenzen beeinflusst.

Grundlage der Ausbildung von Modellgruppen ist das verwendete Abstandsmaß. In der vorliegenden Arbeit werden Modellabstände primär basierend auf den dem Erkennungssystem zur Verfügung stehenden Mustervektoren berechnet. Eine Analyse des Zusammenhangs zwischen Struktur der Merkmalsvektoren und der stochastischen Modellierung zeigt eine hochgradige Korrelation zwischen dem geglätteten akustischen Score und der lokalen Sprechgeschwindigkeit. Es konnte gezeigt werden, dass die Abhängigkeit primär durch die Einführung der Delta-Koeffizienten verursacht wird. Die durch die strukturelle Abhängigkeit verursachte Verschlechterung der Modellierungsgenauigkeit kann durch das Training spezifischer Gruppenmodelle bedingt kompensiert werden. Vielversprechend ist in diesem Zusammenhang das Konzept der Sprechgeschwindigkeitsnormierung, das jedoch nicht Teil dieser Arbeit ist. Die Abhängigkeit kann jedoch vorteilhafterweise zur Bestimmung der aktuell vorliegenden Sprechgeschwindigkeit genutzt werden - eine Klassifikationseinheit basierend auf zusätzlichen Merkmalen ist nicht notwendig.

Die nächsthöhere Ebene der Modellgruppierung wird erreicht, wenn komplette Modellsätze zu Gruppen zusammengefasst werden. Dies ist insbesondere für die Anpassung des Systems an unterschiedliche Sprecher von Bedeutung. Die zugrundeliegende Idee ist, statt eines sprecherunabhängigen Modellsatzes für alle Sprecher individuelle Modellsätze für charakteristische Sprechergruppen zu trainieren. Die Gruppen sollten jedoch automatisch, ohne weiteres Expertenwissen gefunden werden. Untersucht wurden in diesem Zusammenhang insbesondere automatische Techniken zur Modellgruppierung, sog. Clusterverfahren. Diese iterativen Verfahren gruppieren die Daten - hier Modelle - anhand eines vorgegebenen Abstandsmaßes. Als Sprechermodelle wurden hier schwerpunktmäßig die sog. Gauss'schen-Mixtur-Modelle (GMM) untersucht, die für Sprecheridentifizierungssysteme die derzeit leistungsfähigste Modellstruktur darstellen.

In Experimenten wurde die Modellierungsqualität der verschiedenen Sprechermodellstrukturen anhand der Identifizierungsrate bewertet. Hierbei zeigte sich, dass neben der Modellgröße (Anzahl Verteilungen) insbesondere das verwendete Trainingsverfahren von ausschlaggebender Bedeutung ist. In diesem Zusammenhang wurde mit dem Eigenvoice-Ansatz ein neuartiger Trainingsalgorithmus untersucht, der das Wissen über die Varianz zwischen den Sprechermodellen für eine robuste Parameterschätzung ausnutzt. Es konnte gezeigt werden, dass durch eine geschickte Modellstrukturierung mittels des Eigenvoice-Ansatzes eine dem MAP-Training nahezu ebenbürtige Performanz bzgl. der Sprecheridentifizierung erzielt werden kann.

Darüber hinaus ist die dem Eigenvoice-Ansatz zugrundeliegende PCA-Transformation geeignet, den Raum der Modellparameter in einem niedrig-dimensionalen Subraum zu transformieren, in dem eine robuste Sprechergruppierung möglich wird. Die Sprechermodelle sind in diesem Subraum auf sehr wenige Parameter (<100) reduziert, die die relevante Information tragen. Ungenügend geschätzte Komponenten der Sprechermodelle (Ausreißer) können

auf diese Art wirkungsvoll kompensiert werden. Die PCA-Transformation sortiert die Basisvektoren des reduzierten Raums anhand der Varianz in der jeweiligen Raumrichtung. Unter Verwendung einer Euklidischen Distanzmetrik werden die komponentenweisen Abstände der Sprecher im reduzierten Raum daher - indirekt - anhand der Inter-Sprecher Varianz bewertet.

Zusammenfassend lässt sich sagen, dass die Ausbildung von Gruppen einen grundlegenden Mechanismus darstellt, um einen Kompromiss zwischen scharfer Modellierung einerseits, und sicherer Parameterschätzung andererseits, zu erzielen. Die Gruppenbildung selbst kann auf verschiedenen Ebenen der akustisch-phonetischen Modellierung realisiert werden. So lassen sich Verteilungen, (Sub-)Modelle oder auch ganze Modellsätze zu Gruppen zusammenfassen. Durch die gezielte Einbeziehung von Sprecher- bzw. Sprechgeschwindigkeitsinformation in den Gruppierungsprozess, sowie eine robuste Modellselektion in der Erkennungsphase, kann eine deutliche Reduktion der Wortfehlerrate erreicht werden. Insbesondere für eine unüberwachte Adaption der Systemparameter wird hierdurch eine verbesserte Ausgangsbasis geschaffen.

Anhang A

Nomenklatur

A.1 Allgemeine Bedeutung

N, K	Anzahl
v, ROS	Sprechgeschwindigkeit
T	Zeitdauer in Frames oder Sekunden
p	Wahrscheinlichkeit bzw. Wahrscheinlichkeitsdichte
L	Likelihood $\log(p)$, sowie Diskriminanzmaß
S	dito., als Score
$\boldsymbol{\mathcal{X}}$	Menge/Sequenz von Merkmalsvektoren
$\boldsymbol{w}, c$	Gewichtsfaktor(en)
Δ	Unterschied, Abstand
D	Sprecherdistanz
G	Distanz zwischen Sprechergruppen

A.2 Spezielle Variablen

N_B	Zahl der Blatt(=Terminal)knoten in einem Entscheidungsbaum
N_{Bf}	Gesamtzahl der Normalverteilungen
N_F	Fensterbreite
N_I	Anzahl der Phonemklassen
N_K	Gesamtzahl der Knoten in einem Baum
N_L	Anzahl der Schichten (Layer) in einem Entscheidungsbaum
N_M	Anzahl der Modelle
N_P	Anzahl der Mustervektoren (Pattern)
N_{Ph}	Zahl der Phoneme
N_R	Gesamtzahl der Trainingsäußerungen
N_S	Gesamtzahl aller Zustände (States) aller HMMs
K_{EV}	Anzahl der Eigenvoices
K_i	Anzahl der Verteilungen in einem Codebuch

K_R	Anzahl der Regressionsklassen bei MLLR
K_S	Anzahl der Sprecher
K_G	Anzahl der Sprechergruppen

A.3 Abkürzungen

BF	Basisfunktion (Prototyp)
CART	Classification-And-Regression-Tree (Entscheidungsbaum)
EM	Expectation Maximization
EMR	Emission Ratio (Emissionsverhältnis)
EV	Eigenvoices
GD	Gender Dependent (geschlechtsabhängig)
GI	Gender Independent (geschlechsunabhängig)
GMM	Gaussian Mixture Model
HMM	Hidden Markov Model
ICA	Independent Components Analysis
LBG	Linde-Buzo-Gray Algorithmus
LDA	Linear Discriminant Analysis
LR	Linear Regression
LM	Language Model (Sprachmodell)
MAP	Maximum Aposteriori
ML	Maximum Likelihood
MLLR	Maximum Likelihood Linear Regression
MSE	Mean Square Error
NV	Normalverteilung (als Prototyp)
PCA	Principal Components Analysis
ROS	Rate of Speech (Sprechgeschwindigkeit)
SR	Speaking Rate (Sprechgeschwindigkeit, insbes. als Kategorie)
SA	Speaker Adapted (sprecherangepaßt)
SD	Speaker Dependent (sprecherabhängig)
SI	Speaker Independent (sprecherunabhängig)
VFS	Vector Field Smoothing
VQ	Vector Quantization
WDF	Wahrscheinlichkeitsdichtefunktion

Literaturverzeichnis

[Auc99] R. Auckenthaler, E. Parris, M. Carey, "Improving a GMM Speaker Verification System by Phonetic Weighting", Proc. ICASSP, Paper Nr. 1440, 1999.

[Bah91] L. Bahl, P. de Souza, P. Gopalakrishnan, D. Nahamoo, M. Picheny, "Decision Trees for Phonological Rules in Continuous Speech", Proc. ICASSP, S. 185–188, 1991.

[Beh95a] M. Beham, "Merkmalsextraktion und Regelgewinnung für die automatische Spracherkennung", Dissertation, TU München, Lehrstuhl für Mensch-Maschine-Kommunikation, 1995.

[Beh95b] M. Beham, G. Ruske, "Adaptiver stochastischer Sprach/Pause-Detektor", Tagungsband 17. DAGM-Symposium, S. 60–67, 1995.

[Bes98] L. Besacier, J. Bonastre, "Frame Pruning for Speaker Recognition", Proc. ICASSP, S. 765–768, 1998.

[Beu99] K. Beulen, "Phonetische Entscheidungsbäume für die automatische Spracherkennung mit großem Vokabular", Dissertation, RWTH Aachen, Lehrstuhl für Informatik VI, 1999.

[Boc01] E. Bocchieri, B. K. Mak, "Subspace Distribution Clustering Hidden Markov Model", IEEE Trans. on Speech and Audio Processing, Band 9(3), S. 264–275, 2001.

[Bot00] H. Botterweck, "Very Fast Adaptation for Large Vocabulary Continuous Speech Recognition Using Eigenvoices", Proc. ICSLP, Paper Nr. 934, 2000.

[ChK00] K. Chen, W. Liau, H. Wang, L. Lee, "Fast Speaker Adaptation Using Eigenspace-Based Maximum Likelihood Linear Regression", Proc. ICSLP, Paper Nr. 1411, 2000.

[ChS98] S. Chen, P. Gopalakrishnan, "Clustering via the Bayesian Information Criterion with Applications in Speech Recognition", Proc. ICASSP, 645-648, 1998.

[ChR97] R. Chengalvarayan, L. Deng, "Face-Recognition Using View-Based and Modular Eigenspaces", IEEE Trans. on Speech and Audio Processing, Band 5(3), S. 243–256, 1997.

[ChC97] C. Chesta, P. Laface, F. Ravera, "Bottom-Up and Top-Down State Clustering for Robust Acoustic Modeling", Proc. Eurospeech, S. 11–14, 1997.

[Cho99] W. Chou, W. Reichl, "Decision Tree State Tying Based on Penalized Bayesian Information Criterion", Proc. ICASSP, S. 345–348, 1999.

[Col96] J. Colombi, D. Ruck, S. Rogers, M. Oxley, T. Anderson, "Cohort Selection and Word Grammar Effects for Speaker Recognition", Proc. ICASSP, S. 85–88, 1996.

[Dav80] S. Davis, P. Mermelstein, "Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences", IEEE Trans. on Acoustics, Speech, and Signal Processing, Band 28(4), S. 357–366, 1980.

[Dig95b] V. Digalakis, L. Neumeyer, "Speaker Adaptation Using Combined Transformation and Bayesian Methods", Proc. ICASSP, S. 680–683, 1995.

[Dig95a] V. Digalakis, D. Rtischev, L. Neumeyer, "Speaker Adaptation Using Constrained Estimation of Gaussian Mixtures", IEEE Trans. on Speech and Audio Processing, Band 3(5), S. 357–366, 1995.

[Duc97] J. Duchateau, K. Demuynck, D. V. Compernolle, "A Novel Node Splitting Criterion in Decision Tree Construction for Semi-Continuous HMMs", Proc. Eurospeech, S. 1183–1187, 1997.

[Eat94] J. Eatock, J. Mason, "A Quantitative Assessment of the Relative Speaker Discriminating Properties of Phonemes", Proc. ICASSP, S. 133–136, 1994.

[Fab97] T. Fabian, "Implementierung von Methoden der Sprecheradaption in einem automatischen Spracherkennungssystem", Diplomarbeit, TU München, Lehrstuhl für Mensch-Maschine-Kommunikation, 1997.

[Fab01] T. Fabian, T. Pfau, G. Ruske, "Analysis of N-Best Output Hypothesis for Fast Speech in Large Vocabulary Continuous Speech Recognition", Proc. Eurospeech, S. 2535–2538, 2001.

[Fal99] R. Faltlhauser, T. Pfau, G. Ruske, "Creating Hidden Markov Models for Fast Speech by Optimized Clustering", Proc. Eurospeech, S. 407–410, 1999.

[Fal00b] R. Faltlhauser, T. Pfau, G. Ruske, "On-Line Speaking Rate Estimation Using a GMM/NN Approach", Tagungsband ITG-Fachtagung Sprachkommunikation, S. 101–105, 2000.

[Fal00a] R. Faltlhauser, T. Pfau, G. Ruske, "On-Line Speaking Rate Estimation Using Gaussian Mixture Models", Proc. ICASSP, S. 1355–1358, 2000.

[Fal00c] R. Faltlhauser, T. Pfau, G. Ruske, "On the Use of Speaking Rate as a Generalized Feature to Improve Decision Trees", Proc. ICSLP, S. 317–320, 2000.

[Fal01a] R. Faltlhauser, G. Ruske, "Improving Speaker Recognition Performance Using Phonetically Structured Gaussian Mixture Models", Proc. Eurospeech, S. 751–754, 2001.

[Fal01b] R. Faltlhauser, G. Ruske, "Robust Speaker Clustering in Eigenspace", IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), Paper Nr. 86, 2001.

[Fin97a] M. Finke, I. Rogina, "Wide Context Acoustic Modeling in Read vs. Spontaneous Speech", Proc. ICASSP, S. 1743–1746, 1997.

[Fin97b] M. Finke, A. Waibel, "Speaking Mode Dependent Pronunciation Modeling in Large Vocabulary Conversational Speech Recognition", Proc. Eurospeech, S. 2379–2382, 1997.

[Fis97] J. Fiscus, "A Post-Processing System to Yield Reduced Word Error Rates: Recogniser Output Voting Error Reduction (ROVER)", IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), S. 347–354, 1997.

[Foo94] J. Foote, H. Silverman, "A Model Distance Measure for Talker Clustering and Identification", Proc. ICASSP, S. 317–320, 1994.

[For73] G. Forney, "The Viterbi Algorithm", Proc. of the IEEE, Band 61, S. 268–278, 1973.

[Fra00] H. Frank, "Kontextabhängige Modellierung mit Hilfe von Triphonen: Implementierung eines Bottom-Up-Clusterverfahrens", Diplomarbeit, TU München, Lehrstuhl für Mensch-Maschine-Kommunikation, 2000.

[Fue00] C. Fügen, I. Rogina, "Integrating Dynamic Speech Modalities into Context Decision Trees", Proc. ICASSP, S. 1277–1280, 2000.

[Fur89] S. Furui, "Unsupervised Speaker Adaptation Method Based on Hierarchical Spectral Clustering", Proc. ICASSP, S. 286–289, 1989.

[Gal00] M. Gales, "Cluster Adaptive Training of Hidden Markov Models", IEEE Trans. on Speech and Audio Processing, Band 8(4), S. 417–427, 2000.

[Gao97] Y. Gao, M. Padmanabhan, M. Picheny, "Speaker Adaptation Based on Pre-Clustering Training Speakers", Proc. Eurospeech, S. 2091–2094, 1997.

[Gau92] J.-L. Gauvain, C.-H. Lee, "Improved Acoustic Modeling with Bayesian Learning", Proc. ICASSP, S. 481–484, 1992.

[Haz00] T. Hazen, "A Comparison of Novel Techniques for Rapid Speaker Adaptation", Speech Communication, Band 6(31), S. 15–33, 2000.

[He99a] J. He, L. Liu, "Speaker Verification Performance and the Length of Test Sequence", Proc. ICASSP, Paper Nr. 1021, 1999.

[He97] J. He, L. Liu, G. Palm, "A New Codebook Training Algorithm for VQ-Based Speaker Recognition", Proc. ICASSP, S. 1091–1094, 1997.

[He99b] J. He, L. Liu, G. Palm, "A Discriminative Training Algorithm for VQ-Based Speaker Identification", IEEE Trans. on Speech and Audio Processing, Band 7(7), S. 353–356, 1999.

[Hec97] L. Heck, A. Sankar, "Acoustic Clustering and Adaptation for Robust Speech Recognition", Proc. Eurospeech, S. 1867–1870, 1997.

[Hua91] X. Huang, K. Lee, "On Speaker-Independent, Speaker-Dependent, and Speaker-Adaptive Speech Recognition", Proc. ICASSP, S. 877–880, 1991.

[Hwa92] M.-Y. Hwang, X. Huang, "Subphonetic Modeling with Markov States - Senone", Proc. ICASSP, S. 33–36, 1992.

[Ima91] A. Imamura, "Speaker-Adaptive HMM-Based Speech Recognition with a Stochastic Speaker Classifier", Proc. ICASSP, S. 841–844, 1991.

[Imp00] B. Imperl, Z. Kacic, B. Horvat, A. Zgank, "Agglomerative vs. Tree-Based Clustering for the Definition of Multilingual Set of Triphones", Proc. ICASSP, S. 1273–1276, 2000.

[Iso99] T. Isobe, J. Takahashi, "A New Cohort Normalization Using Local Acoustic Information for Speaker Verification", Proc. ICASSP, Paper Nr. 1893, 1999.

[Joh98] S. Johnson, P. Woodland, "Speaker Clustering Using Direct Maximization of the MLLR-Adapted Likelihood", Proc. ICASSP, S. 1775–1778, 1998.

[Jua85] B.-H. Juang, L. Rabiner, "A Probabalistic Distance Measure for Hidden Markov Models", AT&T Technical Journal, Band 64(2), S. 391–408, 1985.

[Jua90] B.-H. Juang, L. Rabiner, "The Segmental K-Means Algorithm for Estimating Parameters of Hidden Markov Models", IEEE Trans. on Acoustics, Speech, and Signal Processing, Band 38(9), S. 1639–1641, 1990.

[Kan00] S. Kanthak, K. Schütz, H. Ney, "Using SIMD Instructions for Fast Likelihood Calculation in LVCSR", Proc. ICASSP, S. 1531–1534, 2000.

[Kem95] T. Kemp, "Data-Driven Codebook Adaptation in Phonetically Tied SCHMMs", Proc. ICASSP, S. 477–479, 1995.

[Kem00] T. Kemp, M. Schmidt, M. Westphal, A. Waibel, "Strategies for Automatic Segmentation of Audio Data", Proc. ICASSP, S. 1423–1426, 2000.

[Kla98] D. Klakow, "Log-Linear Interpolation of Language Models", Proc. ICSLP, S. 1695–1698, 1998.

[Kos94b] T. Kosaka, S. Matsunaga, S. Sagayama, "Tree-Structured Speaker Clustering for Speaker-Independent Continuous Speech Recognition", Proc. ICSLP, S. 1375–1378, 1994.

[Kos94a] T. Kosaka, S. Sagayama, "Tree-Structured Speaker Clustering for Fast Speaker Adaptation", Proc. ICASSP, S. 245–248, 1994.

[Kuh00] R. Kuhn, J.-C. Junqua, P. Nguyen, N. Niedzielski, "Rapid Speaker Adaptation in Eigenvoice Space", IEEE Trans. on Speech and Audio Processing, Band 8(6), S. 695–707, 2000.

[Kuh99] R. Kuhn, P. Nguyen, J. Junqua, R. Boman, N. Niedzielski, S. Fincke, K. Field, M. Contolini, "Fast Speaker Adaptation Using A Priori Knowledge", Proc. ICASSP, Paper Nr. 1587, 1999.

[Kuh98] R. Kuhn, P. Nguyen, J.-C. Junqua, L. Goldwasser, N. Niedzielski, S. Fincke, K. Field, M. Contolini, "Eigenvoices for Speaker Adaptation", Proc. ICSLP, S. 1771–1774, 1998.

[Kuw96] H. Kuwabara, "Acoustic Properties of Phonemes in Continuous Speech for Different Speaking Rate", Proc. ICSLP, S. 2435–2438, 1996.

[Kuw97] H. Kuwabara, "Acoustic and Perceptual Properties of Phonemes in Continuous Speech as a Function of Speaking Rate", Proc. Eurospeech, S. 1003–1006, 1997.

[Laz96] A. Lazarides, Y. Normandin, R. Kuhn, "Improving Decision Trees for Acoustic Modeling", Proc. ICSLP, S. 1053–1057, 1996.

[LeC93] C.-H. Lee, J.-L. Gauvain, "Speaker Adaptation Based on MAP Estimation of HMM Parameters", Proc. ICASSP, S. 558–561, 1993.

[LeL96] L. Lee, R. Rose, "Speaker Normalization using Efficient Frequency Warping Procedures", Proc. ICASSP, S. 353–356, 1996.

[Leg95] C. Leggetter, "Improved Acoustic Modelling for HMMs using Linear Transformations", Dissertation, University of Cambridge, Engineering Department, 1995.

[Lin80] Y. Linde, A. Buzo, R. Gray, "An Algorithm for Vector Quantizer Design", IEEE Trans. on Communications, Band 28(1), S. 84–95, 1980.

[Lue00] A. Lübke, "Konfidenzmaße in der automatischen Spracherkennung", Diplomarbeit, TU München, Lehrstuhl für Mensch-Maschine-Kommunikation, 2000.

[Mak96] B. Mak, E. Barnard, "Phone Clustering using the Bhattacharyya Distance", Proc. ICSLP, Paper Nr. 281, 1996.

[Dud90] M. Mangold, Dudenredaktion, Herausgeber, DUDEN, Aussprachewörterbuch, Dudenverlag, 6. Auflage, 1990.

[Mar98] F. Martinez, D. Tapias, J. Alvarez, "Towards Speech Rate Independence in Large Vocabulary Continuous Speech Recognition", Proc. ICASSP, S. 725–728, 1998.

[Mar97] F. Martinez, D. Tapias, J. Alvarez, P. Leon, "Characteristics of Slow, Average and Fast Speech and Their Effects in Large Vocabulary Continuous Speech Recognition", Proc. Eurospeech, S. 469–472, 1997.

[Mir95] N. Mirghafori, E. Fosler, N. Morgan, "Fast Speakers in Large Vocabulary Continuous Speech Recognition: Analysis & Antidotes", Proc. Eurospeech, S. 491–494, 1995.

[Mir96] N. Mirghafori, E. Fosler, N. Morgan, "Towards Robustness to Fast Speech in ASR", Proc. ICASSP, S. 335–338, 1996.

[Mor97] N. Morgan, E. Fosler, N. Mirghafori, "Speech Recognition using On-Line Estimation of Speaking Rate", Proc. Eurospeech, S. 2079–2082, 1997.

[Mor98] N. Morgan, E. Fosler-Lussier, "Combining Multiple Estimators of Speaking Rate", Proc. ICASSP, S. 729–733, 1991.

[Nae01] C. Naeger, "Implementierung eines diskriminativen Clusterverfahrens für ein HMM-basiertes automatisches Spracherkennungssystem", Technischer Bericht, Lehrstuhl für Mensch-Maschine-Kommunikation, 2001.

[Nai98] M. Naito, L. Deng, Y. Sagisaka, "Speaker Clustering for Speech Recognition using the Parameters Characterizing Vocal-Tract Dimensions", Proc. ICASSP, S. 981–984, 1998.

[Nak97] S. Nakagawa, K. Markov, "Speaker Verification Using Frame and Utterance Level Likelihood Normalization", Proc. ICASSP, S. 1087–1090, 1997.

[Ned01] J. Nedel, R. Stern, "Duration Normalization for Improved Recognition of Spontaneous and Read Speech Via Missing Feature Methods", Proc. ICASSP, S. 313–316, 2001.

[Ngu99] P. Nguyen, C. Wellekens, J.-C. Junqua, "Maximum Likelihood Eigenspace and MLLR for Speech Recognition in Noisy Environments", Proc. Eurospeech, S. 2519–2522, 1999.

[Noc97] H. Nock, M. Gales, S. Young, "A Comparative Study of Methods for Phonetic Decision-Tree State Clustering", Proc. Eurospeech, S. 111–114, 1997.

[Pad98] M. Padmanabhan, L. Bahl, D. Nahamoo, M. Picheny, "Speaker Clustering and Transformation for Speaker Adaptation in Speech Recognition Systems", IEEE Trans. on Speech and Audio Processing, Band 6(1), S. 71–77, 1998.

[Par94] E. Parris, M. Carey, "Discriminative Phonemes for Speaker Identification", Proc. ICSLP, S. 1843–1846, 1994.

[Pau97] D. Paul, "Extensions to Phone-State Decision-Tree Clustering: Single Tree and Tagged Clustering", Proc. ICASSP, S. 1487–1490, 1997.

[Pen94] A. Pentland, B. Moghaddam, T. Starner, "View-Based and Modular Eigenspaces for Face-Recognition", Proc. Conf. on Computer Vision and Pattern Recognition, S. 84–91, 1994.

[Pfa00b] T. Pfau, "Methoden zur Erhöhung der Robustheit automatischer Spracherkennungssysteme gegenüber Variationen der Sprechgeschwindigkeit", Dissertation, TU München, Lehrstuhl für Mensch-Maschine-Kommunikation, 2000.

[Pfa99] T. Pfau, R. Faltlhauser, G. Ruske, "Speaker Normalization and Pronunciation Variant Modeling: Helpful Methods for Improving Recognition of Fast Speech", Proc. Eurospeech, S. 299–302, 1999.

[Pfa00a] T. Pfau, R. Faltlhauser, G. Ruske, "A Combination of Speaker Normalization and Speech Rate Normalization for Automatic Speech Recognition", Proc. ICSLP, S. 362–365, 2000.

[Pfa98b] T. Pfau, G. Ruske, "Creating Hidden-Markov-Models for Fast Speech", Proc. ICSLP, S. 205–208, 1998.

[Pfa98a] T. Pfau, G. Ruske, "Estimating the Speaking Rate By Vowel Detection", Proc. ICASSP, S. 945–948, 1998.

[Pfi96] H. Pfitzinger, "Two Approaches to Speech Rate Estimation", Proc. SST, S. 421–426, 1996.

[Pla95] B. Plannerer, "Erkennung fließender Sprache mit integrierten Suchmethoden", Dissertation, TU München, Lehrstuhl für Mensch-Maschine-Kommunikation, 1995.

[Rab89] L. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", Proceedings of the IEEE, Band 77(2), S. 257–286, 1989.

[Rab86] L. Rabiner, B.-H. Juang, "An Introduction to Hidden Markov Models", IEEE Acoustics, Speech, and Signal Processing Magazine, Band 3(1), S. 4–16, 1986.

[ReW96] W. Reichl, "Diskriminative Lernverfahren für die automatische Spracherkennung", Dissertation, TU München, Lehrstuhl für Mensch-Maschine-Kommunikation, 1996.

[ReW99] W. Reichl, W. Chou, "A Unified Approach of Incorporating General Features in Decision Tree Based Acoustic Modeling", Proc. ICASSP, S. 573–576, 1999.

[ReJ96] J. Reinecke, "Evaluierung der signalnahen Spracherkennung im Verbundprojekt VERBMOBIL (Herbst 1996)", MEMO 113, TU Braunschweig, 1996.

[Rey97] D. Reynolds, "Comparison of Background Normalization Methods for Text-Independent Speaker Verification", Proc. Eurospeech, S. 963–966, 1997.

[Rey95] D. Reynolds, R. Rose, "Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models", IEEE Trans. on Speech and Audio Processing, Band 3(1), S. 72–83, 1995.

[Ric99] M. Richardson, M. Hwang, X. Huang, "Improvements on Speech Recognition for Fast Talkers", Proc. Eurospeech, S. 411–414, 1999.

[Rog97] I. Rogina, "Automatic Architecture Design by Likelihood-Based Context Clustering with Crossvalidation", Proc. Eurospeech, S. 1223–1226, 1997.

[Ros96] A. Rosenberg, S. Parthasarathy, "Speaker Background Models for Connected Digit Password Speaker Verification", Proc. ICASSP, S. 81–84, 1991.

[Rus94] G. Ruske, Automatische Spracherkennung, Oldenburg Verlag, 2. Auflage, 1994.

[Sam98] K. Samudravijaya, S. Singh, P. Rao, "Pre-Recognition Measures of Speaking Rate", Speech Communication, Band 24, S. 73–84, 1998.

[Sch89] O. Schmidbauer, "Ein System zur Lauterkennung in fliessender Sprache auf der Basis artikulatorischer Merkmale", Dissertation, TU München, Lehrstuhl für Datenverarbeitung, 1989.

[Sie95] M. Siegler, R. Stern, "On the Effects of Speech Rate in Large Vocabulary Speech Recognition Systems", Proc. ICASSP, S. 612–615, 1995.

[Six00] A. Sixtus, S. Molau, S. Kanthak, R. Schlüter, H. Ney, "Recent Improvements of the RWTH Large Vocabulary Speech Recognition System on Spontaneous Speech", Proc. ICASSP, S. 1671–1674, 2000.

[Sol98] H. Soltau, "On the Influence of Hyperarticulated Speech on Recognition Performance", Proc. ICSLP, S. 225–228, 1998.

[Sol00] H. Soltau, A. Waibel, "Specialized Acoustic Models for Hyperarticulated Speech", Proc. ICASSP, S. 1779–1782, 2000.

[Tak95] J. Takahashi, S. Sagayama, "Vector-Field-Smoothed Bayesian Learning for Incremental Speaker Adaptation", Proc. ICASSP, S. 696–699, 1995.

[Thy00] O. Thyes, R. Kuhn, P. Nguyen, J.-C. Junqua, "Speaker Identification and Verification Using Eigenvoices", Proc. ICSLP, Paper Nr. 1155, 2000.

[Tsa01] Y. Tsao, S.-M. Lee, L.-S. Lee, "Segmental Eigenvoice for Rapid Speaker Adaptation", Proc. Eurospeech, S. 1269–1272, 2001.

[Tsu00] S. Tsuge, T. Fukada, K. Kita, "Frame-Period Adaptation for Speaking Rate Robust Speech Recognition", Proc. ICSLP, Paper Nr. 283, 2000.

[Tur91a] M. Turk, A. Pentland, "Eigenfaces for Recognition", Journal of Cognitive Neuroscience, Band 3(1), S. 71–86, 1991.

[Tur91b] M. Turk, A. Pentland, "Eigenfaces for Recognition", Proc. Conf. on Computer Vision and Pattern Recognition, S. 586–591, 1991.

[Ver96] J. Verhasselt, J. Martens, "A Fast and Reliable Rate of Speech Detector", Proc. ICSLP, S. 2158–2261, 1996.

[Vog75] A. Vogel, "Ein gemeinsames Funktionsschema zur Beschreibung der Lautheit und der Rauhigkeit", Biol. Cybernetics, Band 18, S. 31–40, 1975.

[Wan01] N. Wang, W.-H. Tsai, L.-S. Lee, "Eigen-MLLR Coefficients as New Feature Parameters for Speaker Identification", Proc. Eurospeech, S. 1385–1388, 2001.

[Wol97] F. Wolfertstetter, "Verallgemeinerte stochastische Modellierung für die automatische Spracherkennung", Dissertation, TU München, Lehrstuhl für Mensch-Maschine-Kommunikation, 1997.

[Wre01] B. Wrede, G. Fink, G. Sagerer, "An Investigation of Modelling Aspects for Rate-Dependent Speech Recognition", Proc. Eurospeech, S. 2527–2530, 2001.

[You94] S. Young, J. Odell, P. Woodland, "Tree-Based State Tying for High Accuracy Acoustic Modelling", Proc. ARPA Workshop on Human Language Technology, S. 307–312, 1994.

[Zhe00] J. Zheng, H. Franco, F. Weng, A. Sankar, H. Bratt, "Word-Level Rate of Speech Modeling using Rate-Specific Phones and Pronunciations", Proc. ICASSP, S. 1775–1778, 2000.