Automatische Topologiegenerierung für kontinuierliche Hidden-Markov-Modelle

Robert Faltlhauser, Günther Ruske
Technische Universität München
Arcisstr. 21, 80290 München
{fal,rus}@mmk.e-technik.tu-muenchen.de

Zusammenfassung

Im allgemeinen wird in Spracherkennungssystemen die Struktur der verwendeten Hidden-Markov-Modelle durch den Designer des Systems von vornherein festgelegt. Im Prinzip wird dadurch jedoch die Topologie der HMMs unabhängig vom verwendeten Trainingsmaterial definiert. Im folgenden wird ein Verfahren vorgestellt, das basierend auf den zur Verfügung stehenden Trainingsdatensätzen die Struktur der Modelle (Anzahl der Zustände, Transitionen, Basisfunktionen) automatisch generiert. Das Verfahren wird für CD-Phonemmodelle verwendet. Ausgehend von einer Initialtrainingssequenz, bei der jeder Merkmalsvektor als Initialprototyp eines Zustandes gesehen wird, wird iterativ das übrige Trainingsset mittels dynamischer Programmierung auf diese Startsequenz abgebildet und danach jeweils eine Reduktion der Zustandszahl vorgenommen.

1 Einleitung

Die meisten heutigen Spracherkennungssysteme verwenden Hidden-Markov-Modelle (HMM) zur Klassifikation der gesprochenen Äußerungen.

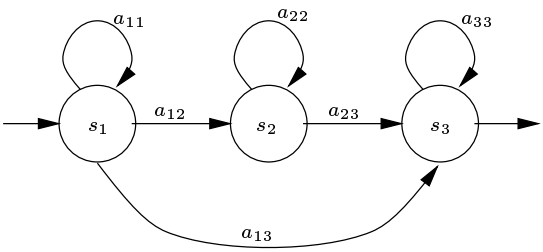

Abbildung 1: Hidden-Markov-Modell mit 3 Zuständen.

Diese Modelle bestehen aus mehreren Zuständen, die von links nach rechts durchlaufen werden. In den einzelnen Zuständen werden mittels multivariater Normalverteilungen (CD-HMM) die Wahrscheinlichkeitsdichtefunktionen $p(\mathbf{x}|i)$ modelliert, wobei $\mathbf{x}$ einen Merkmalsvektor und i einen Zustand eines Modells bezeichnet. Für das Training der Modelle stehen sehr leistungsfähige Lernalgorithmen zur Verfügung: Maximum-Likelihood (ML) oder Maximum-Aposteriori (MAP) im Rahmen stochastischer Lernverfahren oder beispielsweise Minimum-Classification-Error (MCE) bei den diskriminativen Lernverfahren.
Wenig Augenmerk wird allerdings auf die Initialisierung bzw. die grundlegende Struktur der Modelle gelegt. Im allgemeinen wird diese im voraus, zwar unter Beachtung des Trainingsmaterials, aber nichts destoweniger 'manuell' durch den Systemdesigner festgelegt. Bei Verwendung von Phonemen als Grundeinheiten schwanken die Angaben über die Zahl der Zustände der Modelle in der Regel zwischen zwei und sieben. Bei Ganzwortmodellen liegt die Zahl der Zustände noch wesentlich höher.
Aber auch die Anzahl der erlaubten Transitionen wird a-priori festgelegt. So werden teilweise nur die Übergänge von einem Zustand zum nächsten erlaubt. Sinnvoll sind hier aber auch Sprünge unter Auslassung eines oder mehrerer Zwischenzustände.
Im folgenden wird ein Verfahren vorgestellt, das die Parameter (Zustände, Transitionen, Basisfunktionen) während des Initialisierungsvorgangs automatisch bestimmt. In Kapitel 2 werden die grundlegenden Ansätze zur Initialisierung erläutert, die auch im vorgestellten Verfahren zum Zuge kommen, welches dann in Kapitel 3 näher ausgeführt wird. Kernpunkt dieses Verfahrens ist die dynamische Aufteilung der Merkmalsvektoren auf die Zustände. In Kapitel 4 werden dann erste Ergebnisse präsentiert, die mittels dieses Verfahrens erzielt wurden. Im Anschluß daran folgt eine Diskussion der Ergebnisse und ein Ausblick auf noch offene Fragestellungen.

2 Standardverfahren zur Initialisierung

Der grundlegende Ablauf bei der Erstellung von Hidden-Markov-Modellen sieht im allgemeinen folgendermaßen aus:

1. Festlegung der Modelltopologie: Zustände, Transitionen
2. Aufteilung der Merkmalsvektoren auf die Zustände
3. Berechnung der Prototypen: Mittelpunkte, Varianzen

Als erstes wird die Modelltopologie vom Systemdesigner festgelegt. Danach werden die Merkmalsvektoren der Trainingssequenzen meist gleichförmig auf die jeweiligen Zustände aufgeteilt. In Abschnitt 3 wird dagegen ein eigener Ansatz beschrieben, der die gegebenen Vektorsequenzen dynamisch auf die Zustände aufteilt. Im letzten Schritt werden mittels Clusterverfahren die Mittelpunkte/Varianzen der Normalverteilungen aus den zugeordneten Merkmalsvektoren berechnet. Ein bekanntes Clusterverfahren, das auch im Abschnitt 3 vorgestellten Verfahren zum Einsatz kommt, ist der **LBG-Algorithmus** (Linde-Buzo-Gray, [2]). Der Algorithmus besteht aus mehreren Schritten, die iterativ wiederholt werden.

1. Initialprototyp: Mittelpunkt und Varianz der gesamten zugeteilten Merkmalsvektoren.
2. Splitting der Prototypen:
   Jeder mögliche Prototyp $\mathbf{m}_k$ der $\mathbf{K}$ Prototypen eines Zustandes wird in 2 Prototypen $\mathbf{m}_k^1$ und $\mathbf{m}_k^2$ aufgespalten. Bei Verwendung diagonaler Kovarianzmatrizen gilt für jede Dimension:

$$m_{ki}^1 = m_{ki} + 0,5\sigma_{ki} \quad (1)$$
$$m_{ki}^2 = m_{ki} - 0,5\sigma_{ki} \quad (2)$$

3. Neuzuordnung der Vektoren:
   Die Merkmalsvektoren werden den neuen Prototypen zugeordnet: ein Vektor wird dem Prototypen zugeordnet, der nach einem bestimmten Abstandsmaß am nächsten gelegen ist. Im allgemeinen wird hierbei ein gewichteter Euklidischer Abstand verwendet.
4. Neuberechnung der Mittelpunkte:
   Für alle Vektoren, die einem bestimmten Prototypen zugeordnet wurden, wird der Mittelpunkt - und damit der Prototyp - neu bestimmt.

Die Schritte 3 und 4 werden solange wiederholt, bis die relative Änderung ϵ von d (Gl.3) in einer Iteration unter eine gegebene Schwelle ϵ_{min} fällt, oder bis keine Verringerung von d mehr möglich ist. Durch die iterative Wiederholung dieser beiden Schritte wird also der mittlere Abstand

$$d = \sum_{j=1}^{S_V} \min_k (\mathbf{x}_j - \mathbf{m}_k)^T \mathbf{C}^{-1} (\mathbf{x}_j - \mathbf{m}_k) \quad (3)$$

minimiert, wobei $\mathbf{C} = \mathbf{C}_{ges}$ oder $\mathbf{C} = \mathbf{C}_k$.

Die Schritte 2 bis 4 hingegen werden solange wiederholt, bis entweder eine obere Grenze für die Zahl der Prototypen erreicht ist, oder aber keine Prototypen mehr zum Splitten zur Verfügung stehen.

Aber bei all diesen Standardverfahren zur Modellinitialisierung ist der zeitliche Ablauf identisch: die Festlegung der Topologie steht dabei immer an erster Stelle. Die nachfolgende Initialisierung baut dann auf der einmal festgelegten Struktur auf.
Im folgenden wird ein eigener Ansatz erläutert, der zum Ziel hat, die Struktur automatisch während des Initialisierungsvorgangs aus dem zur Verfügung stehenden Trainingsmaterial zu generieren.

3 Dynamische, datenabhängige Ableitung der Modelltopologie

3.1 Initialisierung der Zustände

Um die Topologie der Modelle automatisch zu erzeugen, sind 2 Ansätze möglich:

- Die Modelle werden mit einer minimalen Zustandszahl initialisiert, die bei Bedarf vergrößert wird ("State-splitting").
- Initialisierung mit einer größeren Zustandszahl als nötig; anschließendes Wegstreichen (Pruning) der überflüssigen Zustände.

Das im folgenden näher erläuterte Verfahren basiert auf dem zweiten Ansatz. Bild 2 zeigt die Verteilung der Phonemdauer für das Phonem /e:/. Der Verlauf der Verteilung ist typisch für die meisten Phoneme.

Die im allgemeinen verwendete Zustandszahl für Phonemmodelle liegt im Bereich von 2 bis 10. Bild 2 zeigt, daß die mittlere Phonemdauer ebenfalls in diesen Bereich fällt. Für die Initialisierung wird also eine Länge gewählt, die größer als die mittlere Phonemdauer ist. Die Initialisierung selbst erfolgt mit einer Sequenz des Trainingsmaterials, die die geforderte Länge

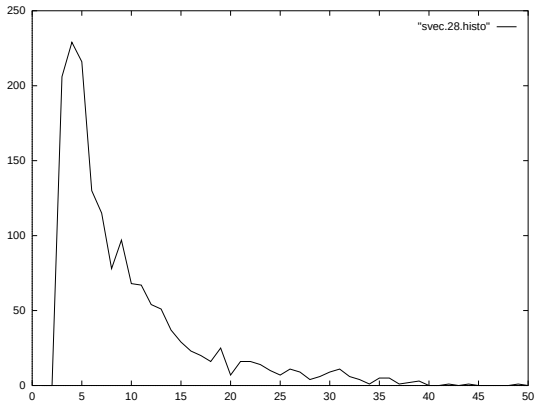

Abbildung 2: Häufigkeitsverteilung der Phonemdauer (in Frames) für Phonem /e:/

aufweist. Jedes einzelne Frame der Sequenz wird dabei jeweils genau einem Zustand zugeordnet [1] und bildet den initialen Prototypen. Einen Extremfall stellt die Initialisierung mit der längsten Sequenz des Trainingsmaterials dar.

3.2 Dynamische Zuteilung der Vektoren

Die Zuordnung einer Sequenz von Merkmalsvektoren zu den Zuständen des Modells erfolgt mittels eines einstufigen Algorithmus der dynamischen Programmierung (DP). Damit wird der mittlere Abstand der Vektoren zu ihren Prototypen minimiert. Als Distanzmaß wird hierbei der gewichtete euklidische Abstand zum nächstgelegenen Prototypen verwendet. Um mit dem in den einzelnen Zuständen ablaufenden Clusteralgorithmus konsistent zu bleiben, wird hier ebenso mit der inversen diagonalen Kovarianzmatrix [2] aller Merkmalsvektoren gewichtet. Als Zustandsübergänge sind in diesem Stadium alle vorwärts gerichteten Transitionen erlaubt, womit alle tatsächlich auftretenden Transitionen erfaßt werden. Aus den aufgetretenen Zustandsübergängen kann die Transitionsmatrix bestimmt werden, die sich wiederum gegebenenfalls an gewisse Erkennereinschränkungen anpassen läßt. Nach dem Backtracking der DP-Matrix erhält man die optimale Zustandssequenz für die gegebene Sequenz von Merkmalsvektoren. Anhand dieser berechneten Zustandssequenz werden die Merkmalsvektoren auf die einzelnen Zustände verteilt.

3.3 Update der Prototypen

Nach der Zuordnung von κ Sequenzen erfolgt die Neuberechnung der Prototypen. Für jeden einzelnen Zustand wird der Clusteralgorithmus (2) gestartet, der aus den dem Zustand zugeordneten Merkmalsvektoren die neuen Prototypen bestimmt. Da ein Update nach Zuordnung einer Sequenz noch keine gravierenden Änderungen an den Prototypen hervorruft und da der Clusteralgorithmus sehr rechenzeitintensiv ist, erfolgt ein Update auch erst nach Zuordnung von κ Sequenzen. κ kann dabei einen festen Wert annehmen, z.B. $\kappa = 10$, oder auch abhängig von der Gesamtzahl $\mathbf{S}_T$ der Trainingssequenzen gesetzt werden: $\kappa = \frac{\mathbf{S}_T}{\kappa_T}$. κ_T gibt hierbei die gewünschte Maximalzahl an Updates an.

3.4 Reduktion der Zustandszahl

Nach Zuordnung aller $\mathbf{S}_T$ Trainingssequenzen erhält man ein Modell, das noch viele unnötige Zustände enthält. Es gilt nun, diese zu entfernen und nur die relevanten Zustände beizubehalten. Für das 'Pruning' der Zustände liegen 2 Ansatzpunkte nahe.

3.4.1 Löschen redundanter Zustände:

Dieser Ansatz berücksichtigt lediglich die Zahl der Vektoren, die einem Zustand zugewiesen wurden. Dazu werden aus allen Zuständen diejenigen mit der größten relativen Häufigkeit bezüglich der Vektorzahl ausgewählt. Die übrigen Zustände werden gelöscht. Das Histogramm über die Vektoranzahl (s. Abb. 3) muß dazu einer Glättung unterworfen werden, da nur die 'wirklichen' Häufigkeitsmaxima gefunden werden sollen.

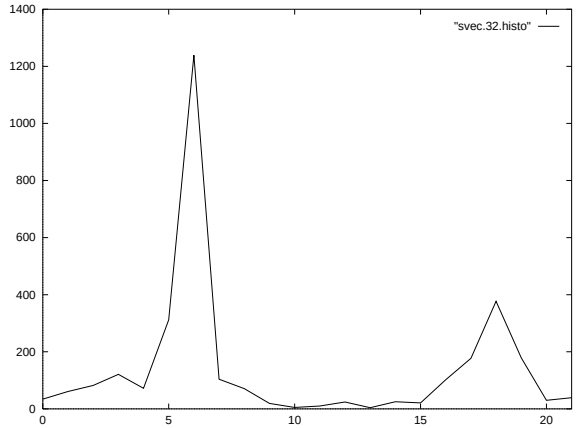

Abbildung 3: Vektoranzahl je Zustand für Phonem /h/

Die Schwankungen, die bei einem diskreten Histogramm bereits als Maxima detektiert würden, werden dadurch eliminiert.

3.4.2 Hierarchisches Vereinigen:

Der zweite Ansatz reduziert iterativ die Zahl der Zustände. 2 benachbarte Zustände werden zu einem vereinigt, wenn gilt:

$$V_g > V_{thres} \quad \text{mit} \quad V_g = \frac{d_i + d_{i+1}}{d_{i,i+1}} \tag{4}$$

wobei d_i der mittlere Abstand der Vektoren eines Zustandes zu den Prototypen ist (Gl.3). d_i und d_{i+1} sind dabei die Abstände der Einzelzustände, wohingegen $d_{i,i+1}$ den Abstand für den vereinigten Zustand angibt.

Die Zustände werden hierarchisch zusammengefaßt: in einer Iteration werden jeweils die beiden benachbarten Zustände vereinigt, die eine maximale Vereinigungsgüte V_g aufweisen. Eine hohe Anzahl an Zuständen gewährleistet die Unterscheidbarkeit der Modelle, da die zeitliche Akzentuierung des Phonems besser modelliert wird. Aus diesem Grund wird in Gl.4 die Schwelle V_{thres} eingeführt, die ein zu starkes Sinken der Zustandszahl verhindern soll. Die Iteration bricht ab, wenn keine benachbarten Zustände mehr existieren, die obige Bedingung erfüllen (Gl.4).

Nach der Zustandsreduktion erfolgt eine Neuzuordnung sämtlicher Trainingssequenzen. Mittels DP werden die Sequenzen auf die verbliebenen Zustände abgebildet. Nach der Zuordnung wird ein letztes Mal in jedem Zustand ein Clusteralgorithmus gestartet, der die finalen Prototypen berechnet.

4 Ergebnisse

Um einen direkten Vergleich der Verfahren mit einem Standardinventar zu ermöglichen, wurde für einen ersten Test mittels 'hierarchischem Vereinigen' ein Phoneminventar (HV) erzeugt, das in Bezug auf die Zustandsanzahl jedes Phonems nahezu dem Standardinventar entspricht. Die Transitionswahrscheinlichkeiten wurden hierbei noch als untrainiert angenommen, d.h. auch das automatisch initialisierte Inventar wird mit einer Gleichverteilung der Transitionswahrscheinlichkeiten belegt. Aus Anpassungsgründen an den verwendeten Erkenner sind die Transitionen vorerst auf 3 mögliche Transitionen beschränkt: Loop, Next und Skip. Das Phoneminventar besteht aus 52 Phonemen einschließlich einem Pause- und einem Garbage-Modell. Tabelle 1 zeigt die mittleren Scores und das Verhältnis des Gesamtscores des korrekten Phonems zum Gesamtscore des jeweiligen besten Konkurrenzphonems.

Da es sich bei den Scores per Definition um logarithmierte Wahrscheinlichkeiten handelt (negative Werte), sollten sie also möglichst nahe bei 0 liegen. Der Scoreanstieg der HV-Modelle in Spalte 1 entspricht demnach einer relativen Verbesserung um 1,8 Prozent. Ein Wert von 1.0 in Spalte 2 würde bedeuten, daß die Konkurrenzphoneme im Mittel den selben Score haben wie das jeweils korrekte Phonem. Aufgrund der negativen Scores zeigt der auf 0.909 gesunkene Wert der HV-Modelle also einen gestiegenen Scoreabstand der korrekten Phoneme zu den Konkurrenzphonemen an.

Inventar	mittl. Score	mittl. Verhältnis
Standard	-55.4	0.916
HV	-54.4	0.909

Tabelle 1: Mittlere Scores des Trainingsmaterials.

5 Diskussion

Die auf diese Weise entstandenen Startmodelle bilden - wie üblich - den Ausgangspunkt für weiterführende Trainingsiterationen. Sie können sowohl für stochastische als auch für diskriminative Lernalgorithmen verwendet werden. Das Ansteigen der Scores macht deutlich, daß die entstandenen Modelle stärker als die Standardmodelle an das Trainingsmaterial angepaßt sind. Dies war eine der Forderungen an die vorgestellten Verfahren. Die erläuterten Ansätze werden zur Entwicklung 'schärferer' Modelle für Sprechgeschwindigkeitsexperimente benötigt. Sollen die neuen Modelle für reine Erkennungsexperimente verwendet werden, so ist u.U. eine Nachbearbeitung nötig. Denn speziell bei geringem Trainingsmaterial besteht die Gefahr, daß die Modelle sich zu stark an das Trainingsmaterial anpassen. Um eine stärkere Generalisierung zu erhalten, können beispielsweise die erhaltenen Varianzen nachträglich manuell aufgeweitet werden.

Literatur

[1] Ph. Lockwood, M. Blanchet: Än Algorithm for the Dynamic Inference of Hidden Markov Models (DIHMM).Proc. ICASSP-93, p. 251-254

[2] F. Wolfertstetter: Verallgemeinerte stochastische Modellierung für die automatische Spracherkennung. Dissertation, Lehrstuhl für Mensch-Maschine-Kommunikation, TU München, 1996.

[3] G. Ruske: Automatische Spracherkennung. Oldenbourg Verlag, 2. Auflage, 1994.