The Application of Hidden Markov Models in Speech Recognition

Documents
Document Sharing Hub

Full document details: IEEE Application of Hidden Markov Models in Speech Recognition

Documents
Document Sharing Hub

Full document details: IEEE Application of Hidden Markov Models in Speech Recognition

The Application of Hidden Markov Models in Speech Recognition

Mark Gales

Cambridge University Engineering Department
Cambridge
CB2 1PZ
UK
mjfg@eng.cam.ac.uk

Steve Young

Cambridge University Engineering Department
Cambridge
CB2 1PZ
UK
sjy@eng.cam.ac.uk

the essence of knowledge

Boston – Delft

Documents
Document Sharing Hub

Full document details: IEEE Application of Hidden Markov Models in Speech Recognition

Foundations and Trends® in Signal Processing

Published, sold and distributed by:
now Publishers Inc.
PO Box 1024
Hanover, MA 02339
USA
Tel. +1-781-985-4510
www.nowpublishers.com
sales@nowpublishers.com

Outside North America:
now Publishers Inc.
PO Box 179
2600 AD Delft
The Netherlands
Tel. +31-6-51115274

The preferred citation for this publication is M. Gales and S. Young, The Application of Hidden Markov Models in Speech Recognition, Foundations and Trends® in Signal Processing, vol 1, no 3, pp 195–304, 2007

ISBN: 978-1-60198-120-2
© 2008 M. Gales and S. Young

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, mechanical, photocopying, recording or otherwise, without prior written permission of the publishers.

Photocopying. In the USA: This journal is registered at the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923. Authorization to photocopy items for internal or personal use, or the internal or personal use of specific clients, is granted by now Publishers Inc for users registered with the Copyright Clearance Center (CCC). The 'services' for users can be found on the internet at: www.copyright.com

For those organizations that have been granted a photocopy license, a separate system of payment has been arranged. Authorization does not extend to other kinds of copying, such as that for general distribution, for advertising or promotional purposes, for creating new collective works, or for resale. In the rest of the world: Permission to photocopy must be obtained from the copyright owner. Please apply to now Publishers Inc., PO Box 1024, Hanover, MA 02339, USA; Tel. +1-781-871-0245; www.nowpublishers.com; sales@nowpublishers.com

now Publishers Inc. has an exclusive license to publish this material worldwide. Permission to use this content must be obtained from the copyright license holder. Please apply to now Publishers, PO Box 179, 2600 AD Delft, The Netherlands, www.nowpublishers.com; e-mail: sales@nowpublishers.com

Documents
Document Sharing Hub

Full document details: IEEE Application of Hidden Markov Models in Speech Recognition

Foundations and Trends® in Signal Processing
Volume 1 Issue 3, 2007

Editorial Board

Editor-in-Chief:
Robert M. Gray
Dept of Electrical Engineering
Stanford University
350 Serra Mall
Stanford, CA 94305
USA
rmgray@stanford.edu

Editors

Abeer Alwan (UCLA)
John Apostolopoulos (HP Labs)
Pamela Cosman (UCSD)
Michelle Effros (California Institute of Technology)
Yonina Eldar (Technion)
Yariv Ephraim (George Mason University)
Sadaoki Furui (Tokyo Institute of Technology)
Vivek Goyal (MIT)
Sinan Gunturk (Courant Institute)
Christine Guillemot (IRISA)
Sheila Hemami (Cornell)
Lina Karam (Arizona State University)
Nick Kingsbury (Cambridge University)
Alex Kot (Nanyang Technical University)
Jelena Kovacevic (CMU)
B.S. Manjunath (UCSB)
Urbashi Mitra (USC)
Thrasos Pappas (Northwestern University)
Mihaela van der Shaar (UCLA)
Luis Torres (Technical University of Catalonia)
Michael Unser (EPFL)
P.P. Vaidyanathan (California Institute of Technology)
Rabab Ward (University of British Columbia)
Susie Wee (HP Labs)
Clifford J. Weinstein (MIT Lincoln Laboratories)
Min Wu (University of Maryland)
Josiane Zerubia (INRIA)

Documents
Document Sharing Hub
Full document details: IEEE Application of Hidden Markov Models in Speech Recognition

Editorial Scope

Foundations and Trends® in Signal Processing will publish survey and tutorial articles on the foundations, algorithms, methods, and applications of signal processing including the following topics:

- Adaptive signal processing
- Audio signal processing
- Biological and biomedical signal processing
- Complexity in signal processing
- Digital and multirate signal processing
- Distributed and network signal processing
- Image and video processing
- Linear and nonlinear filtering
- Multidimensional signal processing
- Multimodal signal processing
- Multiresolution signal processing
- Nonlinear signal processing
- Randomized algorithms in signal processing
- Sensor and multiple source signal processing, source separation
- Signal decompositions, subband and transform methods, sparse representations
- Signal processing for communications
- Signal processing for security and forensic analysis, biometric signal processing
- Signal quantization, sampling, analog-to-digital conversion, coding and compression
- Signal reconstruction, digital-to-analog conversion, enhancement, decoding and inverse problems
- Speech/audio/image/video compression
- Speech and spoken language processing
- Statistical/machine learning
- Statistical signal processing
- Classification and detection
- Estimation and regression
- Tree-structured methods

Information for Librarians

Foundations and Trends® in Signal Processing, 2007, Volume 1, 4 issues. ISSN paper version 1932-8346. ISSN online version 1932-8354. Also available as a combined paper and online subscription.

Documents
Document Sharing Hub

Full document details: IEEE Application of Hidden Markov Models in Speech Recognition

Foundations and Trends® in
Signal Processing
Vol. 1, No. 3 (2007) 195–304
© 2008 M. Gales and S. Young
DOI: 10.1561/2000000004

now
the essence of knowledge

The Application of Hidden Markov Models in Speech Recognition

Mark Gales[1] and Steve Young[2]

[1] *Cambridge University Engineering Department, Trumpington Street, Cambridge, CB2 1PZ, UK, mjfg@eng.cam.ac.uk*
[2] *Cambridge University Engineering Department, Trumpington Street, Cambridge, CB2 1PZ, UK, sjy@eng.cam.ac.uk*

Abstract

Hidden Markov Models (HMMs) provide a simple and effective framework for modelling time-varying spectral vector sequences. As a consequence, almost all present day large vocabulary continuous speech recognition (LVCSR) systems are based on HMMs.

Whereas the basic principles underlying HMM-based LVCSR are rather straightforward, the approximations and simplifying assumptions involved in a direct implementation of these principles would result in a system which has poor accuracy and unacceptable sensitivity to changes in operating environment. Thus, the practical application of HMMs in modern systems involves considerable sophistication.

The aim of this review is first to present the core architecture of a HMM-based LVCSR system and then describe the various refinements which are needed to achieve state-of-the-art performance. These

Documents
Document Sharing Hub

Full document details: IEEE Application of Hidden Markov Models in Speech Recognition

refinements include feature projection, improved covariance modelling, discriminative parameter estimation, adaptation and normalisation, noise compensation and multi-pass system combination. The review concludes with a case study of LVCSR for Broadcast News and Conversation transcription in order to illustrate the techniques described.

Documents
Document Sharing Hub
Full document details: IEEE Application of Hidden Markov Models in Speech Recognition

Contents

1 Introduction **1**

2 Architecture of an HMM-Based Recogniser **5**

2.1 Feature Extraction 6
2.2 HMM Acoustic Models (Basic-Single Component) 8
2.3 N-gram Language Models 15
2.4 Decoding and Lattice Generation 17

3 HMM Structure Refinements **21**

3.1 Dynamic Bayesian Networks 21
3.2 Gaussian Mixture Models 23
3.3 Feature Projections 25
3.4 Covariance Modelling 29
3.5 HMM Duration Modelling 32
3.6 HMMs for Speech Generation 33

4 Parameter Estimation **37**

4.1 Discriminative Training 38
4.2 Implementation Issues 42
4.3 Lightly Supervised and Unsupervised Training 44

Documents
Document Sharing Hub

Full document details: IEEE Application of Hidden Markov Models in Speech Recognition

5 Adaptation and Normalisation **47**

5.1 Feature-Based Schemes 48
5.2 Linear Transform-Based Schemes 51
5.3 Gender/Cluster Dependent Models 55
5.4 Maximum a Posteriori (MAP) Adaptation 57
5.5 Adaptive Training 59

6 Noise Robustness **63**

6.1 Feature Enhancement 65
6.2 Model-Based Compensation 67
6.3 Uncertainty-Based Approaches 71

7 Multi-Pass Recognition Architectures **77**

7.1 Example Architecture 77
7.2 System Combination 79
7.3 Complementary System Generation 81
7.4 Example Application — Broadcast News Transcription 82

Conclusions **93**

Acknowledgments **95**

Notations and Acronyms **97**

References **101**

Documents
Document Sharing Hub

Full document details: IEEE Application of Hidden Markov Models in Speech Recognition

1

Introduction

Automatic continuous speech recognition (CSR) has many potential applications including command and control, dictation, transcription of recorded speech, searching audio documents and interactive spoken dialogues. The core of all speech recognition systems consists of a set of statistical models representing the various sounds of the language to be recognised. Since speech has temporal structure and can be encoded as a sequence of spectral vectors spanning the audio frequency range, the hidden Markov model (HMM) provides a natural framework for constructing such models [13].

HMMs lie at the heart of virtually all modern speech recognition systems and although the basic framework has not changed significantly in the last decade or more, the detailed modelling techniques developed within this framework have evolved to a state of considerable sophistication (e.g. [40, 117, 163]). The result has been steady and significant progress and it is the aim of this review to describe the main techniques by which this has been achieved.

The foundations of modern HMM-based continuous speech recognition technology were laid down in the 1970's by groups at Carnegie-Mellon and IBM who introduced the use of discrete density HMMs

Documents
Document Sharing Hub
Full document details: IEEE Application of Hidden Markov Models in Speech Recognition

[11, 77, 108], and then later at Bell Labs [80, 81, 99] where continuous density HMMs were introduced.[1] An excellent tutorial covering the basic HMM technologies developed in this period is given in [141].

Reflecting the computational power of the time, initial development in the 1980's focussed on either discrete word speaker dependent large vocabulary systems (e.g. [78]) or whole word small vocabulary speaker independent applications (e.g. [142]). In the early 90's, attention switched to continuous speaker-independent recognition. Starting with the artificial 1000 word *Resource Management* task [140], the technology developed rapidly and by the mid-1990's, reasonable accuracy was being achieved for unrestricted speaker independent dictation. Much of this development was driven by a series of DARPA and NSA programmes [188] which set ever more challenging tasks culminating most recently in systems for multilingual transcription of broadcast news programmes [134] and for spontaneous telephone conversations [62].

Many research groups have contributed to this progress, and each will typically have its own architectural perspective. For the sake of logical coherence, the presentation given here is somewhat biassed towards the architecture developed at Cambridge University and supported by the HTK Software Toolkit [189].[2]

The review is organised as follows. Firstly, in *Architecture of a HMM-Based Recogniser* the key architectural ideas of a typical HMM-based recogniser are described. The intention here is to present an overall system design using very basic acoustic models. In particular, simple single Gaussian diagonal covariance HMMs are assumed. The following section *HMM Structure Refinements* then describes the various ways in which the limitations of these basic HMMs can be overcome, for example by transforming features and using more complex HMM output distributions. A key benefit of the statistical approach to speech recognition is that the required models are trained automatically on data.

[1] This very brief historical perspective is far from complete and out of necessity omits many other important contributions to the early years of HMM-based speech recognition.

[2] Available for free download at `htk.eng.cam.ac.uk`. This includes a recipe for building a state-of-the-art recogniser for the Resource Management task which illustrates a number of the approaches described in this review.

Documents
Document Sharing Hub
Full document details: IEEE Application of Hidden Markov Models in Speech Recognition

The section *Parameter Estimation* discusses the different objective functions that can be optimised in training and their effects on performance. Any system designed to work reliably in real-world applications must be robust to changes in speaker and the environment. The section on *Adaptation and Normalisation* presents a variety of generic techniques for achieving robustness. The following section *Noise Robustness* then discusses more specialised techniques for specifically handling additive and convolutional noise. The section *Multi-Pass Recognition Architectures* returns to the topic of the overall system architecture and explains how multiple passes over the speech signal using different model combinations can be exploited to further improve performance. This final section also describes some actual systems built for transcribing English, Mandarin and Arabic in order to illustrate the various techniques discussed in the review. The review concludes in *Conclusions* with some general observations and conclusions.

Documents
Document Sharing Hub
Full document details: IEEE Application of Hidden Markov Models in Speech Recognition

Documents
Document Sharing Hub

Full document details: IEEE Application of Hidden Markov Models in Speech Recognition

2

Architecture of an HMM-Based Recogniser

The principal components of a large vocabulary continuous speech recogniser are illustrated in Figure 2.1. The input audio waveform from a microphone is converted into a sequence of fixed size acoustic vectors $\boldsymbol{Y}_{1:T} = \boldsymbol{y}_1, \ldots, \boldsymbol{y}_T$ in a process called feature extraction. The decoder then attempts to find the sequence of words $\boldsymbol{w}_{1:L} = w_1, \ldots, w_L$ which is most likely to have generated $\boldsymbol{Y}$, i.e. the decoder tries to find

$$\hat{\boldsymbol{w}} = \arg\max_{\boldsymbol{w}} \{P(\boldsymbol{w}|\boldsymbol{Y})\}. \tag{2.1}$$

However, since $P(\boldsymbol{w}|\boldsymbol{Y})$ is difficult to model directly,[1] Bayes' Rule is used to transform (2.1) into the equivalent problem of finding:

$$\hat{\boldsymbol{w}} = \arg\max_{\boldsymbol{w}} \{p(\boldsymbol{Y}|\boldsymbol{w})P(\boldsymbol{w})\}. \tag{2.2}$$

The likelihood $p(\boldsymbol{Y}|\boldsymbol{w})$ is determined by an *acoustic model* and the prior $P(\boldsymbol{w})$ is determined by a *language model*.[2] The basic unit of sound

[1] There are some systems that are based on discriminative models [54] where $P(\boldsymbol{w}|\boldsymbol{Y})$ is modelled directly, rather than using generative models, such as HMMs, where the observation sequence is modelled, $p(\boldsymbol{Y}|\boldsymbol{w})$.

[2] In practice, the acoustic model is not normalised and the language model is often scaled by an empirically determined constant and a word insertion penalty is added i.e., in the log domain the total likelihood is calculated as $\log p(\boldsymbol{Y}|\boldsymbol{w}) + \alpha \log(P(\boldsymbol{w})) + \beta|\boldsymbol{w}|$ where α is typically in the range 8–20 and β is typically in the range $0--20$.

Documents
Document Sharing Hub
Full document details: IEEE Application of Hidden Markov Models in Speech Recognition

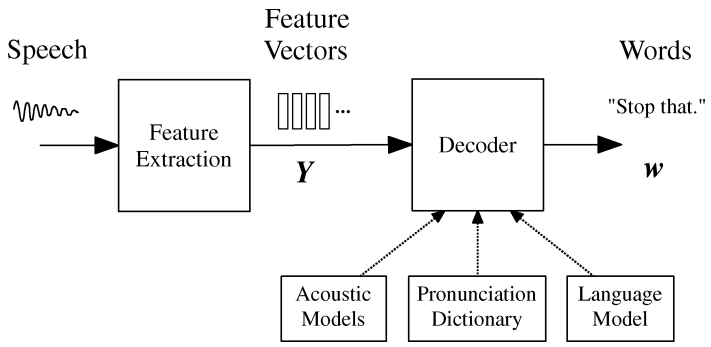

Fig. 2.1 Architecture of a HMM-based Recogniser.

represented by the acoustic model is the *phone*. For example, the word "bat" is composed of three phones /b/ /ae/ /t/. About 40 such phones are required for English.

For any given $\boldsymbol{w}$, the corresponding acoustic model is synthesised by concatenating phone models to make words as defined by a pronunciation dictionary. The parameters of these phone models are estimated from training data consisting of speech waveforms and their orthographic transcriptions. The language model is typically an N-gram model in which the probability of each word is conditioned only on its $N-1$ predecessors. The N-gram parameters are estimated by counting N-tuples in appropriate text corpora. The decoder operates by searching through all possible word sequences using pruning to remove unlikely hypotheses thereby keeping the search tractable. When the end of the utterance is reached, the most likely word sequence is output. Alternatively, modern decoders can generate lattices containing a compact representation of the most likely hypotheses.

The following sections describe these processes and components in more detail.

2.1 Feature Extraction

The feature extraction stage seeks to provide a compact representation of the speech waveform. This form should minimise the loss of

Documents
Document Sharing Hub
Full document details: IEEE Application of Hidden Markov Models in Speech Recognition

information that discriminates between words, and provide a good match with the distributional assumptions made by the acoustic models. For example, if diagonal covariance Gaussian distributions are used for the state-output distributions then the features should be designed to be Gaussian and uncorrelated.

Feature vectors are typically computed every 10 ms using an overlapping analysis window of around 25 ms. One of the simplest and most widely used encoding schemes is based on *mel-frequency cepstral coefficients* (*MFCCs*) [32]. These are generated by applying a truncated discrete cosine transformation (DCT) to a log spectral estimate computed by smoothing an FFT with around 20 frequency bins distributed non-linearly across the speech spectrum. The non-linear frequency scale used is called a *mel scale* and it approximates the response of the human ear. The DCT is applied in order to smooth the spectral estimate and approximately decorrelate the feature elements. After the cosine transform the first element represents the average of the log-energy of the frequency bins. This is sometimes replaced by the log-energy of the frame, or removed completely.

Further psychoacoustic constraints are incorporated into a related encoding called *perceptual linear prediction* (*PLP*) [74]. PLP computes linear prediction coefficients from a perceptually weighted non-linearly compressed power spectrum and then transforms the linear prediction coefficients to cepstral coefficients. In practice, PLP can give small improvements over MFCCs, especially in noisy environments and hence it is the preferred encoding for many systems [185].

In addition to the spectral coefficients, first order (delta) and second-order (delta–delta) regression coefficients are often appended in a heuristic attempt to compensate for the conditional independence assumption made by the HMM-based acoustic models [47]. If the original (static) feature vector is $\boldsymbol{y}_t^{\mathsf{s}}$, then the delta parameter, $\Delta\boldsymbol{y}_t^{\mathsf{s}}$, is given by

$$\Delta\boldsymbol{y}_t^{\mathsf{s}} = \frac{\sum_{i=1}^{n} w_i \left(\boldsymbol{y}_{t+i}^{\mathsf{s}} - \boldsymbol{y}_{t-i}^{\mathsf{s}}\right)}{2\sum_{i=1}^{n} w_i^2} \tag{2.3}$$

Documents
Document Sharing Hub
Full document details: IEEE Application of Hidden Markov Models in Speech Recognition

where n is the window width and w_i are the regression coefficients.[3] The delta–delta parameters, $\Delta^2 \boldsymbol{y}_t^{\mathsf{s}}$, are derived in the same fashion, but using differences of the delta parameters. When concatenated together these form the feature vector $\boldsymbol{y}_t$,

$$\boldsymbol{y}_t = \begin{bmatrix} \boldsymbol{y}_t^{\mathsf{s}\mathsf{T}} & \Delta \boldsymbol{y}_t^{\mathsf{s}\mathsf{T}} & \Delta^2 \boldsymbol{y}_t^{\mathsf{s}\mathsf{T}} \end{bmatrix}^{\mathsf{T}}. \tag{2.4}$$

The final result is a feature vector whose dimensionality is typically around 40 and which has been partially but not fully decorrelated.

2.2 HMM Acoustic Models (Basic-Single Component)

As noted above, each spoken word w is decomposed into a sequence of K_w basic sounds called *base phones*. This sequence is called its pronunciation $\boldsymbol{q}_{1:K_w}^{(w)} = q_1, \ldots, q_{K_w}$. To allow for the possibility of multiple pronunciations, the likelihood $p(\boldsymbol{Y}|\boldsymbol{w})$ can be computed over multiple pronunciations[4]

$$p(\boldsymbol{Y}|\boldsymbol{w}) = \sum_{\boldsymbol{Q}} p(\boldsymbol{Y}|\boldsymbol{Q}) P(\boldsymbol{Q}|\boldsymbol{w}), \tag{2.5}$$

where the summation is over all valid pronunciation sequences for $\boldsymbol{w}$, $\boldsymbol{Q}$ is a particular sequence of pronunciations,

$$P(\boldsymbol{Q}|\boldsymbol{w}) = \prod_{l=1}^{L} P(\boldsymbol{q}^{(w_l)}|w_l), \tag{2.6}$$

and where each $\boldsymbol{q}^{(w_l)}$ is a valid pronunciation for word w_l. In practice, there will only be a very small number of alternative pronunciations for each w_l making the summation in (2.5) easily tractable.

Each base phone q is represented by a continuous density HMM of the form illustrated in Figure 2.2 with transition probability parameters $\{a_{ij}\}$ and output observation distributions $\{b_j()\}$. In operation, an HMM makes a transition from its current state to one of its connected states every time step. The probability of making a particular

[3] In HTK to ensure that the same number of frames is maintained after adding delta and delta–delta parameters, the start and end elements are replicated to fill the regression window.

[4] Recognisers often approximate this by a *max* operation so that alternative pronunciations can be decoded as though they were alternative word hypotheses.

Documents
Document Sharing Hub
Full document details: IEEE Application of Hidden Markov Models in Speech Recognition

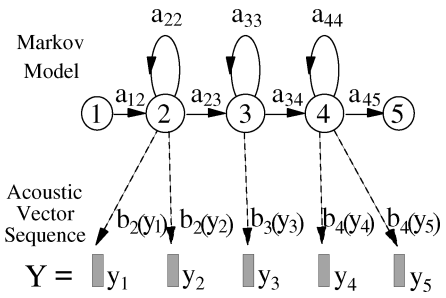

Fig. 2.2 HMM-based phone model.

transition from state $\mathtt{s}_i$ to state $\mathtt{s}_j$ is given by the transition probability $\{a_{ij}\}$. On entering a state, a feature vector is generated using the distribution associated with the state being entered, $\{b_j()\}$. This form of process yields the standard conditional independence assumptions for an HMM:

- states are conditionally independent of all other states given the previous state;
- observations are conditionally independent of all other observations given the state that generated it.

For a more detailed discussion of the operation of an HMM see [141].

For now, single multivariate Gaussians will be assumed for the output distribution:

$$b_j(\boldsymbol{y}) = \mathcal{N}(\boldsymbol{y}; \boldsymbol{\mu}^{(j)}, \boldsymbol{\Sigma}^{(j)}), \tag{2.7}$$

where $\boldsymbol{\mu}^{(j)}$ is the mean of state $\mathtt{s}_j$ and $\boldsymbol{\Sigma}^{(j)}$ is its covariance. Since the dimensionality of the acoustic vector $\boldsymbol{y}$ is relatively high, the covariances are often constrained to be diagonal. Later in *HMM Structure Refinements*, the benefits of using mixtures of Gaussians will be discussed.

Given the composite HMM $\boldsymbol{Q}$ formed by concatenating all of the constituent base phones $\boldsymbol{q}^{(w_1)}, \ldots, \boldsymbol{q}^{(w_L)}$ then the acoustic likelihood is given by

$$p(\boldsymbol{Y}|\boldsymbol{Q}) = \sum_{\boldsymbol{\theta}} p(\boldsymbol{\theta}, \boldsymbol{Y}|\boldsymbol{Q}), \tag{2.8}$$

Documents
Document Sharing Hub
Full document details: IEEE Application of Hidden Markov Models in Speech Recognition

where $\boldsymbol{\theta} = \theta_0, \ldots, \theta_{T+1}$ is a state sequence through the composite model and

$$p(\boldsymbol{\theta}, \boldsymbol{Y}|\boldsymbol{Q}) = a_{\theta_0\theta_1} \prod_{t=1}^{T} b_{\theta_t}(\boldsymbol{y}_t) a_{\theta_t\theta_{t+1}}. \tag{2.9}$$

In this equation, θ_0 and θ_{T+1} correspond to the non-emitting entry and exit states shown in Figure 2.2. These are included to simplify the process of concatenating phone models to make words. For simplicity in what follows, these non-emitting states will be ignored and the focus will be on the state sequence $\theta_1, \ldots, \theta_T$.

The acoustic model parameters $\boldsymbol{\lambda} = [\{a_{ij}\}, \{b_j()\}]$ can be efficiently estimated from a corpus of training utterances using the forward–backward algorithm [14] which is an example of expectation-maximisation (EM) [33]. For each utterance $\boldsymbol{Y}^{(r)}, r = 1, \ldots, R$, of length $T^{(r)}$ the sequence of baseforms, the HMMs that correspond to the word-sequence in the utterance, is found and the corresponding composite HMM constructed. In the first phase of the algorithm, the *E-step*, the *forward probability* $\alpha_t^{(rj)} = p(\boldsymbol{Y}_{1:t}^{(r)}, \theta_t = \mathtt{s}_j; \boldsymbol{\lambda})$ and the *backward probability* $\beta_t^{(ri)} = p(\boldsymbol{Y}_{t+1:T^{(r)}}^{(r)} | \theta_t = \mathtt{s}_i; \boldsymbol{\lambda})$ are calculated via the following recursions

$$\alpha_t^{(rj)} = \left[\sum_i \alpha_{t-1}^{(ri)} a_{ij} \right] b_j\left(\boldsymbol{y}_t^{(r)}\right) \tag{2.10}$$

$$\beta_t^{(ri)} = \left[\sum_j a_{ij} b_j(\boldsymbol{y}_{t+1}^{(r)}) \beta_{t+1}^{(rj)} \right], \tag{2.11}$$

where i and j are summed over all states. When performing these recursions under-flow can occur for long speech segments, hence in practice the log-probabilities are stored and *log arithmetic* is used to avoid this problem [89].[5]

Given the forward and backward probabilities, the probability of the model occupying state $\mathtt{s}_j$ at time t for any given utterance r is just

$$\gamma_t^{(rj)} = P(\theta_t = \mathtt{s}_j | \boldsymbol{Y}^{(r)}; \boldsymbol{\lambda}) = \frac{1}{P^{(r)}} \alpha_t^{(rj)} \beta_t^{(rj)}, \tag{2.12}$$

[5] This is the technique used in HTK [189].

Documents
Document Sharing Hub
Full document details: IEEE Application of Hidden Markov Models in Speech Recognition

where $P^{(r)} = p(\boldsymbol{Y}^{(r)}; \boldsymbol{\lambda})$. These state *occupation probabilities*, also called *occupation counts*, represent a soft alignment of the model states to the data and it is straightforward to show that the new set of Gaussian parameters defined by [83]

$$\hat{\boldsymbol{\mu}}^{(j)} = \frac{\sum_{r=1}^{R}\sum_{t=1}^{T^{(r)}} \gamma_t^{(rj)} \boldsymbol{y}_t^{(r)}}{\sum_{r=1}^{R}\sum_{t=1}^{T^{(r)}} \gamma_t^{(rj)}} \tag{2.13}$$

$$\hat{\boldsymbol{\Sigma}}^{(j)} = \frac{\sum_{r=1}^{R}\sum_{t=1}^{T^{(r)}} \gamma_t^{(rj)} (\boldsymbol{y}_t^{(r)} - \hat{\boldsymbol{\mu}}^{(j)})(\boldsymbol{y}_t^{(r)} - \hat{\boldsymbol{\mu}}^{(j)})^{\mathsf{T}}}{\sum_{r=1}^{R}\sum_{t=1}^{T^{(r)}} \gamma_t^{(rj)}} \tag{2.14}$$

maximise the likelihood of the data given these alignments. A similar re-estimation equation can be derived for the transition probabilities

$$\hat{a}_{ij} = \frac{\sum_{r=1}^{R} \frac{1}{P^{(r)}} \sum_{t=1}^{T^{(r)}} \alpha_t^{(ri)} a_{ij} b_j(\boldsymbol{y}_{t+1}^{(r)}) \beta_{t+1}^{(rj)} \boldsymbol{y}_t^{(r)}}{\sum_{r=1}^{R}\sum_{t=1}^{T^{(r)}} \gamma_t^{(ri)}}. \tag{2.15}$$

This is the second or *M-step* of the algorithm. Starting from some initial estimate of the parameters, $\boldsymbol{\lambda}^{(0)}$, successive iterations of the EM algorithm yield parameter sets $\boldsymbol{\lambda}^{(1)}, \boldsymbol{\lambda}^{(2)}, \ldots$ which are guaranteed to improve the likelihood up to some local maximum. A common choice for the initial parameter set $\boldsymbol{\lambda}^{(0)}$ is to assign the global mean and covariance of the data to the Gaussian output distributions and to set all transition probabilities to be equal. This gives a so-called *flat start* model.

This approach to acoustic modelling is often referred to as the *beads-on-a-string* model, so-called because all speech utterances are represented by concatenating a sequence of phone models together. The major problem with this is that decomposing each vocabulary word into a sequence of context-independent base phones fails to capture the very large degree of context-dependent variation that exists in real speech. For example, the base form pronunciations for "mood" and "cool" would use the same vowel for "oo," yet in practice the realisations of "oo" in the two contexts are very different due to the influence of the preceding and following consonant. Context independent phone models are referred to as *monophones*.

A simple way to mitigate this problem is to use a unique phone model for every possible pair of left and right neighbours. The resulting

Documents
Document Sharing Hub
Full document details: IEEE Application of Hidden Markov Models in Speech Recognition

models are called *triphones* and if there are N base phones, there are N^3 potential triphones. To avoid the resulting data sparsity problems, the complete set of *logical* triphones L can be mapped to a reduced set of physical models P by clustering and tying together the parameters in each cluster. This mapping process is illustrated in Figure 2.3 and the parameter tying is illustrated in Figure 2.4 where the notation $x-q+y$ denotes the triphone corresponding to phone q spoken in the context of a preceding phone x and a following phone y. Each base

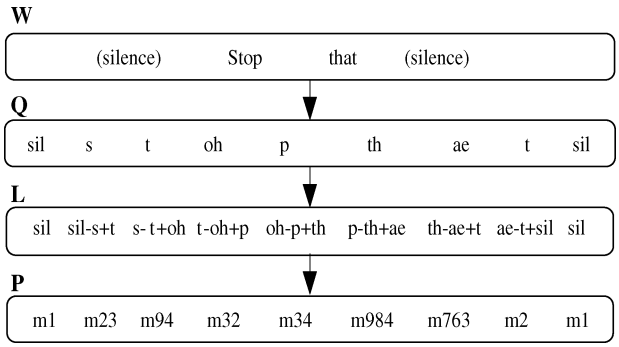

Fig. 2.3 Context dependent phone modelling.

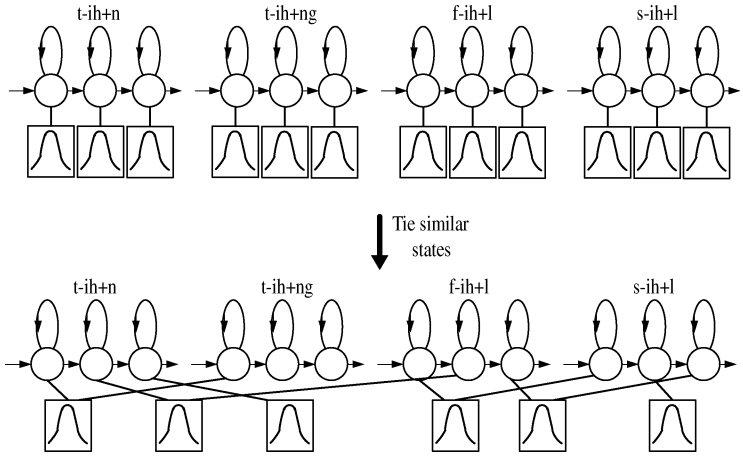

Fig. 2.4 Formation of tied-state phone models.

Documents
Document Sharing Hub
Full document details: IEEE Application of Hidden Markov Models in Speech Recognition

phone pronunciation q is derived by simple look-up from the pronunciation dictionary, these are then mapped to logical phones according to the context, finally the logical phones are mapped to physical models. Notice that the context-dependence spreads across word boundaries and this is essential for capturing many important phonological processes. For example, the /p/ in "stop that" has its burst suppressed by the following consonant.

The clustering of logical to physical models typically operates at the state-level rather than the model level since it is simpler and it allows a larger set of physical models to be robustly estimated. The choice of which states to tie is commonly made using decision trees [190]. Each state position[6] of each phone q has a binary tree associated with it. Each node of the tree carries a question regarding the context. To cluster state i of phone q, all states i of all of the logical models derived from q are collected into a single pool at the root node of the tree. Depending on the answer to the question at each node, the pool of states is successively split until all states have trickled down to leaf nodes. All states in each leaf node are then tied to form a physical model. The questions at each node are selected from a predetermined set to maximise the likelihood of the training data given the final set of state-tyings. If the state output distributions are single component Gaussians and the state occupation counts are known, then the increase in likelihood achieved by splitting the Gaussians at any node can be calculated simply from the counts and model parameters without reference to the training data. Thus, the decision trees can be grown very efficiently using a greedy iterative node splitting algorithm. Figure 2.5 illustrates this tree-based clustering. In the figure, the logical phones s-aw+n and t-aw+n will both be assigned to leaf node 3 and hence they will share the same central state of the representative physical model.[7]

The partitioning of states using phonetically driven decision trees has several advantages. In particular, logical models which are required but were not seen at all in the training data can be easily synthesised. One disadvantage is that the partitioning can be rather coarse. This

[6] Usually each phone model has three states.

[7] The total number of tied-states in a large vocabulary speaker independent system typically ranges between 1000 and 10,000 states.

Documents
Document Sharing Hub
Full document details: IEEE Application of Hidden Markov Models in Speech Recognition

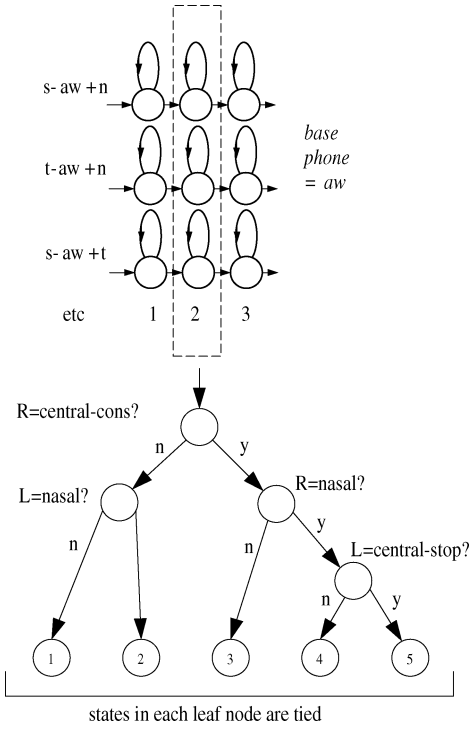

Fig. 2.5 Decision tree clustering.

problem can be reduced using so-called *soft-tying* [109]. In this scheme, a post-processing stage groups each state with its one or two nearest neighbours and pools all of their Gaussians. Thus, the single Gaussian models are converted to mixture Gaussian models whilst holding the total number of Gaussians in the system constant.

To summarise, the core acoustic models of a modern speech recogniser typically consist of a set of tied three-state HMMs with Gaussian output distributions. This core is commonly built in the following steps [189, Ch. 3]:

(1) A flat-start monophone set is created in which each base phone is a monophone single-Gaussian HMM with means and covariances equal to the mean and covariance of the training data.

Documents
Document Sharing Hub
Full document details: IEEE Application of Hidden Markov Models in Speech Recognition

(2) The parameters of the Gaussian monophones are re-estimated using 3 or 4 iterations of EM.
(3) Each single Gaussian monophone q is cloned once for each distinct triphone $x - q + y$ that appears in the training data.
(4) The resulting set of *training-data* triphones is again re-estimated using EM and the state occupation counts of the last iteration are saved.
(5) A decision tree is created for each state in each base phone, the training-data triphones are mapped into a smaller set of tied-state triphones and iteratively re-estimated using EM.

The final result is the required tied-state context-dependent acoustic model set.

2.3 N-gram Language Models

The prior probability of a word sequence $\boldsymbol{w} = w_1, \ldots, w_K$ required in (2.2) is given by

$$P(\boldsymbol{w}) = \prod_{k=1}^{K} P(w_k | w_{k-1}, \ldots, w_1). \tag{2.16}$$

For large vocabulary recognition, the conditioning word history in (2.16) is usually truncated to $N-1$ words to form an *N-gram* language model

$$P(\boldsymbol{w}) = \prod_{k=1}^{K} P(w_k | w_{k-1}, w_{k-2}, \ldots, w_{k-N+1}), \tag{2.17}$$

where N is typically in the range 2–4. Language models are often assessed in terms of their *perplexity*, H, which is defined as

$$\begin{aligned} H &= -\lim_{K\to\infty} \frac{1}{K} \log_2(P(w_1, \ldots, w_K)) \\ &\approx -\frac{1}{K} \sum_{k=1}^{K} \log_2(P(w_k | w_{k-1}, w_{k-2}, \ldots, w_{k-N+1})), \end{aligned}$$

where the approximation is used for N-gram language models with a finite length word sequence.

Documents
Document Sharing Hub
Full document details: IEEE Application of Hidden Markov Models in Speech Recognition

The N-gram probabilities are estimated from training texts by counting N-gram occurrences to form maximum likelihood (ML) parameter estimates. For example, let $C(w_{k-2}w_{k-1}w_k)$ represent the number of occurrences of the three words $w_{k-2}w_{k-1}w_k$ and similarly for $C(w_{k-2}w_{k-1})$, then

$$P(w_k|w_{k-1}, w_{k-2}) \approx \frac{C(w_{k-2}w_{k-1}w_k)}{C(w_{k-2}w_{k-1})}. \tag{2.18}$$

The major problem with this simple ML estimation scheme is data sparsity. This can be mitigated by a combination of discounting and backing-off. For example, using so-called *Katz smoothing* [85]

$$P(w_k|w_{k-1}, w_{k-2}) = \begin{cases} d\frac{C(w_{k-2}w_{k-1}w_k)}{C(w_{k-2}w_{k-1})} & \text{if } 0 < C \leq C' \\ \frac{C(w_{k-2}w_{k-1}w_k)}{C(w_{k-2}w_{k-1})} & \text{if } C > C' \\ \alpha(w_{k-1}, w_{k-2})\ P(w_k|w_{k-1}) & \text{otherwise,} \end{cases} \tag{2.19}$$

where C' is a count threshold, C is short-hand for $C(w_{k-2}w_{k-1}w_k)$, d is a discount coefficient and α is a normalisation constant. Thus, when the N-gram count exceeds some threshold, the ML estimate is used. When the count is small the same ML estimate is used but discounted slightly. The discounted probability mass is then distributed to the unseen N-grams which are approximated by a weighted version of the corresponding bigram. This idea can be applied recursively to estimate any sparse N-gram in terms of a set of back-off weights and $(N-1)$-grams. The discounting coefficient is based on the Turing-Good estimate $d = (r+1)n_{r+1}/rn_r$ where n_r is the number of N-grams that occur exactly r times in the training data. There are many variations on this approach [25]. For example, when training data is very sparse, Kneser–Ney smoothing is particularly effective [127].

An alternative approach to robust language model estimation is to use class-based models in which for every word w_k there is a corresponding class c_k [21, 90]. Then,

$$P(\boldsymbol{w}) = \prod_{k=1}^{K} P(w_k|c_k)p(c_k|c_{k-1}, \ldots, c_{k-N+1}). \tag{2.20}$$

As for word based models, the class N-gram probabilities are estimated using ML but since there are far fewer classes (typically a few hundred)

Documents
Document Sharing Hub
Full document details: IEEE Application of Hidden Markov Models in Speech Recognition

data sparsity is much less of an issue. The classes themselves are chosen to optimise the likelihood of the training set assuming a bigram class model. It can be shown that when a word is moved from one class to another, the change in perplexity depends only on the counts of a relatively small number of bigrams. Hence, an iterative algorithm can be implemented which repeatedly scans through the vocabulary, testing each word to see if moving it to some other class would increase the likelihood [115].

In practice it is found that for reasonably sized training sets,[8] an effective language model for large vocabulary applications consists of a smoothed word-based 3 or 4-gram interpolated with a class-based trigram.

2.4 Decoding and Lattice Generation

As noted in the introduction to this section, the most likely word sequence $\hat{\boldsymbol{w}}$ given a sequence of feature vectors $\boldsymbol{Y}_{1:T}$ is found by searching all possible state sequences arising from all possible word sequences for the sequence which was most likely to have generated the observed data $\boldsymbol{Y}_{1:T}$. An efficient way to solve this problem is to use dynamic programming. Let $\phi_t^{(j)} = \max_{\boldsymbol{\theta}} \{p(\boldsymbol{Y}_{1:t}, \theta_t = \mathtt{s}_j; \boldsymbol{\lambda})\}$, i.e., the maximum probability of observing the partial sequence $\boldsymbol{Y}_{1:t}$ and then being in state $\mathtt{s}_j$ at time t given the model parameters $\boldsymbol{\lambda}$. This probability can be efficiently computed using the Viterbi algorithm [177]

$$\phi_t^{(j)} = \max_i \left\{ \phi_{t-1}^{(i)} a_{ij} \right\} b_j(\boldsymbol{y}_t). \tag{2.21}$$

It is initialised by setting $\phi_0^{(j)}$ to 1 for the initial, non-emitting, entry state and 0 for all other states. The probability of the most likely word sequence is then given by $\max_j\{\phi_T^{(j)}\}$ and if every maximisation decision is recorded, a traceback will yield the required best matching state/word sequence.

In practice, a direct implementation of the Viterbi algorithm becomes unmanageably complex for continuous speech where the topology of the models, the language model constraints and the need to

[8] i.e. $\gg 10^7$ words.

Documents
Document Sharing Hub
Full document details: IEEE Application of Hidden Markov Models in Speech Recognition

bound the computation must all be taken into account. N-gram language models and cross-word triphone contexts are particularly problematic since they greatly expand the search space. To deal with this, a number of different architectural approaches have evolved. For Viterbi decoding, the search space can either be constrained by maintaining multiple hypotheses in parallel [173, 191, 192] or it can be expanded dynamically as the search progresses [7, 69, 130, 132]. Alternatively, a completely different approach can be taken where the breadth-first approach of the Viterbi algorithm is replaced by a depth-first search. This gives rise to a class of recognisers called *stack decoders*. These can be very efficient, however, because they must compare hypotheses of different lengths, their run-time search characteristics can be difficult to control [76, 135]. Finally, recent advances in weighted finite-state transducer technology enable all of the required information (acoustic models, pronunciation, language model probabilities, etc.) to be integrated into a single very large but highly optimised network [122]. This approach offers both flexibility and efficiency and is therefore extremely useful for both research and practical applications.

Although decoders are designed primarily to find the solution to (2.21), in practice, it is relatively simple to generate not just the most likely hypothesis but the N-best set of hypotheses. N is usually in the range 100–1000. This is extremely useful since it allows multiple passes over the data without the computational expense of repeatedly solving (2.21) from scratch. A compact and efficient structure for storing these hypotheses is the *word lattice* [144, 167, 187].

A word lattice consists of a set of nodes representing points in time and a set of spanning arcs representing word hypotheses. An example is shown in Figure 2.6 part (a). In addition to the word IDs shown in the figure, each arc can also carry score information such as the acoustic and language model scores.

Lattices are extremely flexible. For example, they can be rescored by using them as an input recognition network and they can be expanded to allow rescoring by a higher order language model. They can also be compacted into a very efficient representation called a *confusion network* [42, 114]. This is illustrated in Figure 2.6 part (b) where the "-" arc labels indicate null transitions. In a confusion network, the nodes no

Documents
Document Sharing Hub
Full document details: IEEE Application of Hidden Markov Models in Speech Recognition

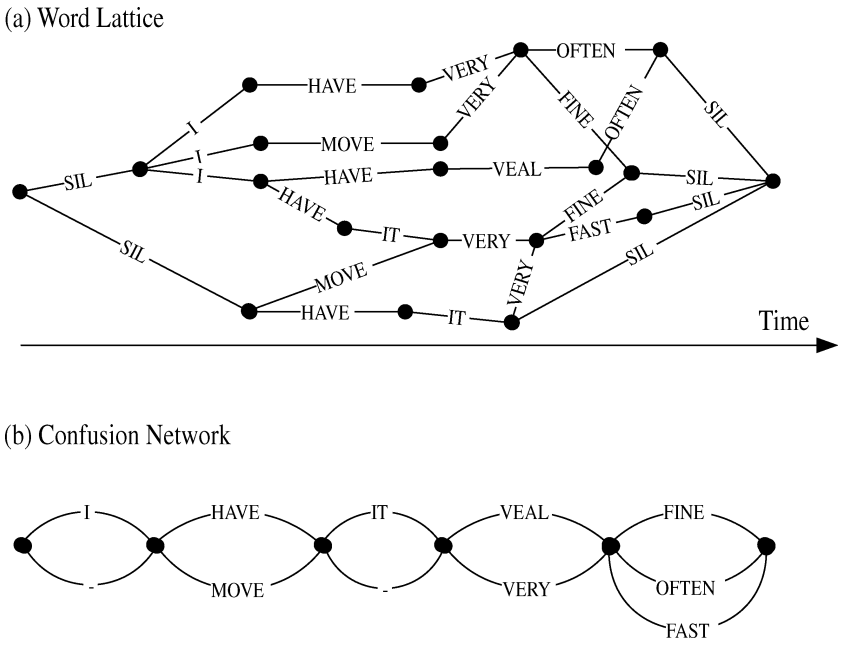

Fig. 2.6 Example lattice and confusion network.

longer correspond to discrete points in time, instead they simply enforce word sequence constraints. Thus, parallel arcs in the confusion network do not necessarily correspond to the same acoustic segment. However, it is assumed that most of the time the overlap is sufficient to enable parallel arcs to be regarded as competing hypotheses. A confusion network has the property that for every path through the original lattice, there exists a corresponding path through the confusion network. Each arc in the confusion network carries the posterior probability of the corresponding word w. This is computed by finding the *link probability* of w in the lattice using a forward–backward procedure, summing over all occurrences of w and then normalising so that all competing word arcs in the confusion network sum to one. Confusion networks can be used for minimum word-error decoding [165] (an example of minimum Bayes' risk (MBR) decoding [22]), to provide confidence scores and for merging the outputs of different decoders [41, 43, 63, 72] (see *Multi-Pass Recognition Architectures*).

Documents
Document Sharing Hub
Full document details: IEEE Application of Hidden Markov Models in Speech Recognition

Documents
Document Sharing Hub

Full document details: IEEE Application of Hidden Markov Models in Speech Recognition

3

HMM Structure Refinements

In the previous section, basic acoustic HMMs and their use in ASR systems have been explained. Although these simple HMMs may be adequate for small vocabulary and similar limited complexity tasks, they do not perform well when used for more complex, and larger vocabulary tasks such as broadcast news transcription and dictation. This section describes some of the extensions that have been used to improve the performance of ASR systems and allow them to be applied to these more interesting and challenging domains.

The use of dynamic Bayesian networks to describe possible extensions is first introduced. Some of these extensions are then discussed in detail. In particular, the use of Gaussian mixture models, efficient covariance models and feature projection schemes are presented. Finally, the use of HMMs for generating, rather than recognising, speech is briefly discussed.

3.1 Dynamic Bayesian Networks

In *Architecture of an HMM-Based Recogniser*, the HMM was described as a generative model which for a typical phone has three emitting

Documents
Document Sharing Hub
Full document details: IEEE Application of Hidden Markov Models in Speech Recognition

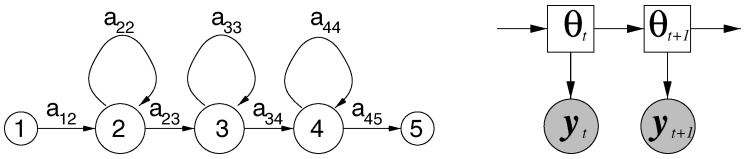

Fig. 3.1 Typical phone HMM topology (left) and dynamic Bayesian network (right).

states, as shown again in Figure 3.1. Also shown in Figure 3.1 is an alternative, complementary, graphical representation called a *dynamic Bayesian network* (*DBN*) which emphasises the conditional dependencies of the model [17, 200] and which is particularly useful for describing a variety of extensions to the basic HMM structure. In the DBN notation used here, squares denote discrete variables; circles continuous variables; shading indicates an observed variable; and no shading an unobserved variable. The lack of an arc between variables shows conditional independence. Thus Figure 3.1 shows that the observations generated by an HMM are conditionally independent given the unobserved, hidden, state that generated it.

One of the desirable attributes of using DBNs is that it is simple to show extensions in terms of how they modify the conditional independence assumptions of the model. There are two approaches, which may be combined, to extend the HMM structure in Figure 3.1: adding additional un-observed variables; and adding additional dependency arcs between variables.

An example of adding additional arcs, in this case between observations is shown in Figure 3.2. Here the observation distribution is

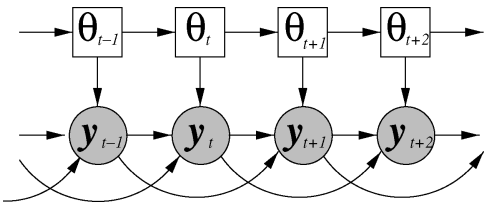

Fig. 3.2 Dynamic Bayesian networks for buried Markov models and HMMs with vector linear predictors.

Documents
Document Sharing Hub
Full document details: IEEE Application of Hidden Markov Models in Speech Recognition

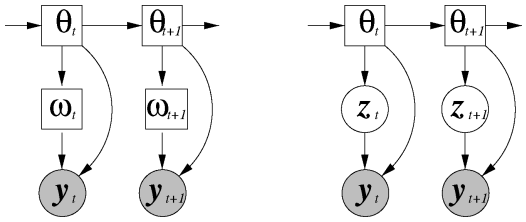

Fig. 3.3 Dynamic Bayesian networks for Gaussian mixture models (left) and factor-analysed models (right).

dependent on the previous two observations in addition to the state that generated it. This DBN describes HMMs with explicit temporal correlation modelling [181], vector predictors [184], and buried Markov models [16]. Although an interesting direction for refining an HMM, this approach has not yet been adopted in mainstream state-of-the-art systems.

Instead of adding arcs, additional unobserved, latent or hidden, variables may be added. To compute probabilities, these hidden variables must then be marginalised out in the same fashion as the unobserved state sequence for HMMs. For continuous variables this requires a continuous integral over all values of the hidden variable and for discrete variables a summation over all values. Two possible forms of latent variable are shown in Figure 3.3. In both cases the temporal dependencies of the model, where the discrete states are conditionally independent of all other states given the previous state, are unaltered allowing the use of standard Viterbi decoding routines. In the first case, the observations are dependent on a discrete latent variable. This is the DBN for an HMM with Gaussian mixture model (GMM) state-output distributions. In the second case, the observations are dependent on a continuous latent variable. This is the DBN for an HMM with factor-analysed covariance matrices. Both of these are described in more detail below.

3.2 Gaussian Mixture Models

One of the most commonly used extensions to standard HMMs is to model the state-output distribution as a mixture model. In *Architecture of an HMM-Based Recogniser*, a single Gaussian distribution was used

Documents
Document Sharing Hub
Full document details: IEEE Application of Hidden Markov Models in Speech Recognition

to model the state–output distribution. This model therefore assumed that the observed feature vectors are symmetric and unimodal. In practice this is seldom the case. For example, speaker, accent and gender differences tend to create multiple modes in the data. To address this problem, the single Gaussian state-output distribution may be replaced by a mixture of Gaussians which is a highly flexible distribution able to model, for example, asymmetric and multi-modal distributed data.

As described in the previous section, mixture models may be viewed as adding an additional discrete latent variable to the system. The likelihood for state $\mathtt{s}_j$ is now obtained by summing over all the component likelihoods weighted by their prior

$$b_j(\boldsymbol{y}) = \sum_{m=1}^{M} c_{jm} \mathcal{N}(\boldsymbol{y}; \boldsymbol{\mu}^{(jm)}, \boldsymbol{\Sigma}^{(jm)}), \tag{3.1}$$

where c_{jm} is the prior probability for component m of state $\mathtt{s}_j$. These priors satisfy the standard constraints for a valid probability mass function (PMF)

$$\sum_{m=1}^{M} c_{jm} = 1, \quad c_{jm} \geq 0. \tag{3.2}$$

This is an example of the general technique of mixture modelling. Each of the M components of the mixture model is a Gaussian probability density function (PDF).

Since an additional latent variable has been added to the acoustic model, the form of the EM algorithm previously described needs to be modified [83]. Rather than considering the complete data-set in terms of state-observation pairings, state/component-observations are used. The estimation of the model parameters then follows the same form as for the single component case. For the mean of Gaussian component m of state $\mathtt{s}_j$, $\mathtt{s}_{jm}$,

$$\hat{\boldsymbol{\mu}}^{(jm)} = \frac{\sum_{r=1}^{R} \sum_{t=1}^{T^{(r)}} \gamma_t^{(rjm)} \boldsymbol{y}_t^{(r)}}{\sum_{r=1}^{R} \sum_{t=1}^{T^{(r)}} \gamma_t^{(rjm)}}, \tag{3.3}$$

where $\gamma_t^{(rjm)} = P(\theta_t = \mathtt{s}_{jm} | \boldsymbol{Y}^{(r)}; \boldsymbol{\lambda})$ is the probability that component m of state $\mathtt{s}_j$ generated the observation at time t of sequence $\boldsymbol{Y}^{(r)}$.

Documents
Document Sharing Hub
Full document details: IEEE Application of Hidden Markov Models in Speech Recognition

R is the number of training sequences. The component priors are estimated in a similar fashion to the transition probabilities.

When using GMMs to model the state-output distribution, variance flooring is often applied.[1] This prevents the variances of the system becoming too small, for example when a component models a very small number of tightly packed observations. This improves generalisation.

Using GMMs increases the computational overhead since when using log-arithmetic, a series of log-additions are required to compute the GMM likelihood. To improve efficiency, only components which provide a "reasonable" contribution to the total likelihood are included. To further decrease the computational load the GMM likelihood can be approximated simply by the maximum over all the components (weighted by the priors).

In addition, it is necessary to determine the number of components per state in the system. A number of approaches, including discriminative schemes, have been adopted for this [27, 106]. The simplest is to use the same number of components for all states in the system and use held-out data to determine the optimal numbers. A popular alternative approach is based on the Bayesian information criterion (BIC) [27]. Another approach is to make the number of components assigned to the state a function of the number of observations assigned to that state [56].

3.3 Feature Projections

In *Architecture for an HMM-Based Recogniser*, dynamic first and second differential parameters, the so-called *delta* and *delta–delta* parameters, were added to the static feature parameters to overcome the limitations of the conditional independence assumption associated with HMMs. Furthermore, the DCT was assumed to approximately decorrelate the feature vector to improve the diagonal covariance approximation and reduce the dimensionality of the feature vector.

It is also possible to use data-driven approaches to decorrelate and reduce the dimensionality of the features. The standard approach is to

[1] See HTK for a specific implementation of variance flooring [189].

Documents
Document Sharing Hub
Full document details: IEEE Application of Hidden Markov Models in Speech Recognition

use linear transformations, that is

$$\boldsymbol{y}_t = \mathbf{A}_{[p]}\tilde{\boldsymbol{y}}_t, \tag{3.4}$$

where $\mathbf{A}_{[p]}$ is a $p \times d$ linear-transform, d is the dimension of the source feature vector $\tilde{\boldsymbol{y}}_t$, and p is the size of the transformed feature vector $\boldsymbol{y}_t$.

The detailed form of a feature vector transformation such as this depends on a number of choices. Firstly, the construction of $\tilde{\boldsymbol{y}}_t$ must be decided and, when required, the class labels that are to be used, for example phone, state or Gaussian component labels. Secondly, the dimensionality p of the projected data must be determined. Lastly, the criterion used to estimate $\mathbf{A}_{[p]}$ must be specified.

An important issue when estimating a projection is whether the class labels of the observations are to be used, or not, i.e., whether the projection should be estimated in a *supervised* or *unsupervised* fashion. In general supervised approaches yield better projections since it is then possible to estimate a transform to improve the discrimination between classes. In contrast, in unsupervised approaches only general attributes of the observations, such as the variances, may be used. However, supervised schemes are usually computationally more expensive than unsupervised schemes. Statistics, such as the means and covariance matrices are needed for each of the class labels, whereas unsupervised schemes effectively only ever have one class.

The features to be projected are normally based on the standard MFCC, or PLP, feature-vectors. However, the treatment of the delta and delta–delta parameters can vary. One approach is to splice neighbouring static vectors together to form a composite vector of typically 9 frames [10]

$$\tilde{\boldsymbol{y}}_t = \begin{bmatrix} \boldsymbol{y}_{t-4}^{\mathsf{s}\mathsf{T}} & \cdots & \boldsymbol{y}_t^{\mathsf{s}\mathsf{T}} & \cdots & \boldsymbol{y}_{t+4}^{\mathsf{s}\mathsf{T}} \end{bmatrix}^{\mathsf{T}}. \tag{3.5}$$

Another approach is to expand the static, delta and delta–delta parameters with third-order dynamic parameters (the difference of delta–deltas) [116]. Both have been used for large vocabulary speech recognition systems. The dimensionality of the projected feature vector is often determined empirically since there is only a single parameter to tune. In [68] a comparison of a number of possible class definitions, phone, state or component, were compared for supervised projection

Documents
Document Sharing Hub
Full document details: IEEE Application of Hidden Markov Models in Speech Recognition

schemes. The choice of class is important as the transform tries to ensure that each of these classes is as separable as possible from all others. The study showed that performance using state and component level classes was similar, and both were better than higher level labels such as phones or words. In practice, the vast majority of systems use component level classes since, as discussed later, this improves the assumption of diagonal covariance matrices.

The simplest form of criterion for estimating the transform is principal component analysis (PCA). This is an unsupervised projection scheme, so no use is made of class labels. The estimation of the PCA transform can be found by finding the p rows of the orthonormal matrix $\mathbf{A}$ that maximises

$$\mathcal{F}_{\texttt{pca}}(\boldsymbol{\lambda}) = \log\left(|\mathbf{A}_{[p]}\tilde{\boldsymbol{\Sigma}}_{\texttt{g}}\mathbf{A}_{[p]}^{\mathsf{T}}|\right), \tag{3.6}$$

where $\tilde{\boldsymbol{\Sigma}}_{\texttt{g}}$ is the total, or global, covariance matrix of the original data, $\tilde{\boldsymbol{y}}_t$. This selects the orthogonal projections of the data that maximises the total variance in the projected subspace. Simply selecting subspaces that yield large variances does not necessarily yield subspaces that discriminate between the classes. To address this problem, supervised approaches such as the linear discriminant analysis (LDA) criterion [46] can be used. In LDA the objective is to increase the ratio of the between class variance to the average within class variance for each dimension. This criterion may be expressed as

$$\mathcal{F}_{\texttt{lda}}(\boldsymbol{\lambda}) = \log\left(\frac{|\mathbf{A}_{[p]}\tilde{\boldsymbol{\Sigma}}_{\texttt{b}}\mathbf{A}_{[p]}^{\mathsf{T}}|}{|\mathbf{A}_{[p]}\tilde{\boldsymbol{\Sigma}}_{\texttt{w}}\mathbf{A}_{[p]}^{\mathsf{T}}|}\right), \tag{3.7}$$

where $\tilde{\boldsymbol{\Sigma}}_{\texttt{b}}$ is the between-class covariance matrix and $\tilde{\boldsymbol{\Sigma}}_{\texttt{w}}$ the average within-class covariance matrix where each distinct Gaussian component is usually assumed to be a separate class. This criterion yields an orthonormal transform such that the average within-class covariance matrix is diagonalised, which should improve the diagonal covariance matrix assumption.

The LDA criterion may be further refined by using the actual class covariance matrices rather than using the averages. One such form is heteroscedastic discriminant analysis (HDA) [149] where the following

Documents
Document Sharing Hub
Full document details: IEEE Application of Hidden Markov Models in Speech Recognition