Chapter 1
Hidden Markov Models and the Variants

Abstract This chapter builds upon the reviews in the previous chapter on aspects of probability theory and statistics including random variables and Gaussian mixture models, and extends the reviews to the Markov chain and the hidden Markov sequence or model (HMM). Central to the HMM is the concept of state, which is itself a random variable typically taking discrete values. Extending from a Markov chain to an HMM involves adding uncertainty or a statistical distribution on each of the states in the Markov chain. Hence, an HMM is a doubly-stochastic process, or probabilistic function of a Markov chain. When the state of the Markov sequence or HMM is confined to be discrete and the distributions associated with the HMM states do not overlap, we reduce it to a Markov chain. This chapter covers several key aspects of the HMM, including its parametric characterization, its simulation by random number generators, its likelihood evaluation, its parameter estimation via the EM algorithm, and its state decoding via the Viterbi algorithm or a dynamic programming procedure. We then provide discussions on the use of the HMM as a generative model for speech feature sequences and its use as the basis for speech recognition. Finally, we discuss the limitations of the HMM, leading to its various extended versions, where each state is made associated with a dynamic system or a hidden time-varying trajectory instead of with a temporally independent stationary distribution such as a Gaussian mixture. These variants of the HMM with state-conditioned dynamic systems expressed in the state-space formulation are introduced as a generative counterpart of the recurrent neural networks to be described in detail in Chapter ??.

1.1 Introduction

In the previous chapter, we reviewed aspects of probability theory and basic statistics, where we introduced the concept of random variables and the asso-

ciated concept of probability distributions. We then discussed Gaussian and mixture-of-Gaussian random variables and their vector-valued or multivariate versions. All these concepts and examples are *static*, meaning that they do not have the temporal dimension making the length or the dimension of the random vectors variable according to how long the temporal sequence is. For a *static* section of the speech signal, its features based on spectral magnitudes (e.g., cepstra) can be well characterized by the multivariate distribution of mixture of Gaussians. This gives rise to the Gaussian mixture model (GMM) of speech features for a short-term or static speech sound pattern.

In this chapter, we will extend the concept of the random variable to the (discrete-time) random sequence, which is a collection of random variables indexed by uniformly spaced discrete times with a variable length. For the general statistical characterization of random sequences, see Chapter 3 of [43] but in this chapter we will extract only the part on Markov sequences as the most commonly used class of general random sequences. The concept of state is essential to a Markov sequence. When the state of the Markov sequence is confined to be discrete, we have a Markov chain, where all possible values taken by the discrete state variable constitutes the (discrete) state space and which we will cover in Section 1.2.

When each discrete state value is generalized to be a new random variable (discrete or continuous), the Markov chain is then generalized to the (discrete or continuous) hidden Markov sequence, or the Hidden Markov Model (HMM) when it is used to characterize or approximate statistical properties of real-world data sequences. In Section 1.3, we first parameterize the HMM in terms of transition probabilities of the underlying Markov chain and of the distributional parameters in the static PDFs given a fixed state. We then show how an HMM can be simulated via probabilistic sampling. Efficient computation of the likelihood of an observation sequence given the HMM is covered in detail, as this is an important but non-obvious element in applying the HMM to speech recognition and other practical problems.

Then, in Section 1.4, we first provide background information on the EM algorithm for maximum likelihood estimation of the parameters in general statistical models that contain hidden or latent random variables. We then apply the EM algorithm to solving the learning or parameter-estimation problem for the HMM (as well as the GMM which can be viewed as a special case of the HMM). The resulting algorithm for the HMM learning is the celebrated Baum-Welch algorithm, widely used in speech recognition and other applications involving the HMM. Step-by-step derivations of the E-step in the Baum-Welch algorithm are given, which provides the conditional probabilities of an HMM state given the input training data. The M-step for the estimation of the transition probabilities of the Markov chain and of the mean vectors and covariance matrices in the Gaussian HMM is also given with step-by-step derivation.

We use Section 3.5 to present the celebrated Viterbi algorithm for optimally decoding the HMM state sequence given the input sequence data. The

technique of dynamic programming, which is the underlying principle of the Viterbi algorithm, is described.

Finally, in Section 3.6, we connect the HMM as a statistical model to practical speech problems. The discussion starts with the HMM's capability as an elegant generative model for speech feature sequences; e.g., [4, 5, 3, 72]. The reasonably good match between the HMM and speech data enables this generative model to be used for the classification task of speech recognition via the use of Bayes rule [41, 57]. An analysis of the weaknesses of the HMM as a generative model for speech motivates its extensions to several variants, where the temporal independence and stationarity in the distribution of the observed speech data conditioned on each HMM state is replaced by more realistic, non-stationary, and temporally correlated dynamic systems with latent or hidden structure [81, 44, 27, 78, 15, 100]. The mathematical formalism of such dynamic systems expressed as the state-space model naturally bridges these HMM variants to the recurrent neural networks to be presented later in Chapter 13 of this book.

1.2 Markov Chains

A Markov chain is a discrete-state Markov sequence, a special case of a general Markov sequence. The state space of a Markov chain is of a discrete nature and is finite: $q_t \in \{s^{(j)}, j = 1, 2, \cdots, N\}$. Each of these discrete values is associated with a state in the Markov chain. Because of the one-to-one correspondence between state $s^{(j)}$ and its index j, we often use the two interchangeably.

A Markov chain,$\mathbf{q}_1^T = q_1, q_2, \cdots, q_T$, is completely characterized by the transition probabilities, defined by

$$P(q_t = s^{(j)} | q_{t-1} = s^{(i)}) \doteq a_{ij}(t), \qquad i, j = 1, 2, \cdots, N \tag{1.1}$$

and by the initial state-distribution probabilities. If these transition probabilities are independent of time t, then we have a homogeneous Markov chain.

The transition probabilities of a (homogeneous) Markov chain are often conveniently put into a matrix form:

$$A = [a_{ij}], \qquad \text{where} \quad a_{ij} \geq 0 \qquad \forall i, j; \quad \text{and} \quad \sum_{j=1}^{N} a_{ij} = 1 \quad \forall i \tag{1.2}$$

which is called the transition matrix of the Markov chain. Given the transition probabilities of a Markov chain, the state-occupation probability

$$p_j(t) \doteq P[q_t = s^{(j)}]$$

can be easily computed. The computation is recursive according to

$$p_i(t+1) = \sum_{j=1}^{N} a_{ji} p_j(t), \quad \forall i. \tag{1.3}$$

If the state-occupation distribution of a Markov chain asymptotically converges: $p_i(t) \to \pi(q^{(i)})$ as $t \to \infty$, we then call $p(s^{(i)})$ a stationary distribution of the Markov chain. For a Markov chain to have a stationary distribution, its transition probabilities, a_{ij}, have to satisfy

$$\pi(s^{(i)}) = \sum_{j=1}^{N} a_{ji} \pi(s^{(j)}), \quad \forall i. \tag{1.4}$$

The stationary distribution of a Markov chain plays an important role in a class of powerful statistical methods collectively named Markov Chain Monte Carlo (MCMC) methods. These methods are used to simulate (i.e., to sample or to draw) arbitrarily complex distributions, enabling one to carry out many difficult statistical inference and learning tasks which would otherwise be mathematically intractable. The theoretical foundation of the MCMC methods is the asymptotic convergence of a Markov chain to its stationary distribution, $\pi(s^{(i)})$. That is, regardless of the initial distribution, the Markov chain is an asymptotically unbiased draw from $\pi(s^{(i)})$. Therefore, in order to sample from an arbitrarily complex distribution, $p(s)$, one can construct a Markov chain, by designing appropriate transition probabilities, a_{ij}, so that its stationary distribution is $\pi(s) = p(s)$.

Three other interesting and useful properties of a Markov chain can be easily derived. First, the state duration in a Markov chain is an exponential or geometric distribution: $p_i(d) = C\,(a_{ii})^{d-1}$, where the normalizing constant is $C = 1 - a_{ii}$. Second, the mean state duration is

$$\overline{d}_i = \sum_{d=1}^{\infty} d\, p_i(d) = \sum_{d=1}^{\infty} (1 - a_{ii})(a_{ii})^{d-1} = \frac{1}{1 - a_{ii}}. \tag{1.5}$$

Finally, the probability for an arbitrary observation sequence of a Markov chain, which is a finite-length state sequence $\mathbf{q}_1^T$, can be easily evaluated. This is simply the product of the transition probabilities traversing the Markov chain: $P(\mathbf{q}_1^T) = \pi_{q_1} \prod_{t=1}^{T-1} a_{q_t q_{t+1}}$, where π_{s_1} is the initial state-occupation probability at $t = 1$.

1.3 Hidden Markov Sequences and Models

Let us view the Markov chain discussed above as an information source capable of generating observational output sequences. Then we can call the Markov chain an observable Markov sequence because its output has one-to-one correspondence to a state. That is, each state corresponds to a deterministically observable variable or event. There is no randomness in the output in any given state. This lack of randomness makes the Markov chain too restrictive to describe many real-world informational sources, such as speech feature sequences, in an adequate manner.

Extension of the Markov chain to embed randomness which overlaps among the states in the Markov chain gives rise to a hidden Markov sequence. This extension is accomplished by associating an observation probability distribution with each state in the Markov chain. The Markov sequence thus defined is a doubly embedded random sequence whose underlying Markov chain is not directly observable, hence a hidden sequence. The underlying Markov chain in the hidden Markov sequence can be observed only through a separate random function characterized by the observation probability distributions.

Note that if the observation probability distributions do not overlap across the states, then the underlying Markov chain would not be hidden. This is because, despite the randomness embedded in the states, any observational value over a fixed range specific to a state would be able to map uniquely to this state. In this case, the hidden Markov sequence essentially reduces to a Markov chain. Some excellent and more detailed exposition on the relationship between a Markov chain and its probabilistic function, or a hidden Markov sequence, can be found in [105, 106].

When a hidden Markov sequence is used to describe a physical, real-world informational source, i.e., to approximate the statistical characteristics of such a source, we often call it a hidden Markov model (HMM). One very successful practical use of the HMM has been in speech processing applications, including speech recognition and its noise robustness, speaker recognition, speech synthesis, and speech enhancement; e.g.,[105, 1, 12, 17, 66, 85, 83, 82, 126, 128, 130, 47, 46, 71, 48, 113, 122]. In these applications, the HMM is used as a powerful model to characterize the temporally nonstationary, spatially variable, but regular, learnable patterns of the speech signal. One key aspect of the HMM as the acoustic model of speech is its sequentially arranged Markov states which permit the use of piecewise stationarity for approximating the globally nonstationary properties of speech feature sequences. Very efficient algorithms have been developed to efficiently optimize the boundaries of the local quasi-stationary temporal regimes, which we will discuss in Section 3.6.

1.3.1 Characterization of a hidden Markov model

We now give a formal characterization of a hidden Markov sequence model or HMM in terms of its basic elements and parameters.

1. Transition probabilities, $\mathbf{A} = [a_{ij}]$, $i, j = 1, 2, ..., N$, of a homogeneous Markov chain with a total of N states

$$a_{ij} = P(q_t = j | q_{t-1} = i), \qquad i, j = 1, 2, \cdots, N. \tag{1.6}$$

2. Initial Markov chain state-occupation probabilities:$\pi = [\pi_i]$, $i = 1, 2, \cdots, N$, where $\pi_i = P(q_1 = i)$.
3. Observation probability distribution, $P(\mathbf{o}_t | s^{(i)}), i = 1, 2, ..., N$. if $\mathbf{o}_t$ is discrete, the distribution associated with each state gives the probabilities of symbolic observations $\{\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_K\}$:

$$b_i(k) = P[\mathbf{o}_t = \mathbf{v}_k | q_t = i], \qquad i = 1, 2, \cdots, N. \tag{1.7}$$

   If the observation probability distribution is continuous, then the parameters, Λ_i, in the PDF characterize state$i$ in the HMM.

The most common and successful PDF used in speech processing for characterizing the continuous observation probability distribution in the HMM is a multivariate mixture Gaussian distribution for vector-valued observation $\mathbf{o}_t \in \mathcal{R}^{\mathcal{D}}$):

$$b_i(\mathbf{o}_t) = \sum_{m=1}^{M} \frac{c_{i,m}}{(2\pi)^{D/2} |\boldsymbol{\Sigma}_{i,m}|^{1/2}} \exp\left[-\frac{1}{2} (\mathbf{o}_t - \boldsymbol{\mu}_{i,m})^{\mathrm{T}} \boldsymbol{\Sigma}_{i,m}^{-1} (\mathbf{o}_t - \boldsymbol{\mu}_{i,m}) \right] \tag{1.8}$$

In this Gaussian-mixture HMM, the parameter set Λ_i comprises scalar mixture weights, $c_{i,m}$, Gaussian mean vectors, $\boldsymbol{\mu}_{i,m} \in \mathcal{R}^{\mathcal{D}}$, and Gaussian covariance matrices, $\boldsymbol{\Sigma}_{i,m} \in \mathcal{R}^{D \times D}$.

When the number of mixture components is reduced to one: $M = 1$, the state-dependent output PDF reverts to a (uni-modal) Gaussian:

$$b_i(\mathbf{o}_t) = \frac{1}{(2\pi)^{D/2} |\boldsymbol{\Sigma}_i|^{1/2}} \exp\left[-\frac{1}{2} (\mathbf{o}_t - \boldsymbol{\mu}_i)^{\mathrm{T}} \boldsymbol{\Sigma}_i^{-1} (\mathbf{o}_t - \boldsymbol{\mu}_i) \right] \tag{1.9}$$

and the corresponding HMM is commonly called a (continuous-density) Gaussian HMM.

Given the model parameters, one convenient way of characterizing a Gaussian HMM is to view it as a generative device producing a sequence of observational data, $\mathbf{o}_t, t = 1, 2, ..., T$. In this view, the data at each time t is generated from the model according to

$$\mathbf{o}_t = \boldsymbol{\mu}_i + \mathbf{r}_t(\boldsymbol{\Sigma}_{\mathrm{i}}), \tag{1.10}$$

where state$i$ at a given time t is determined by the evolution of the Markov chain characterized by a_{ij}, and

$$\mathbf{r}_t(\boldsymbol{\Sigma}_i) = \mathcal{N}(0, \boldsymbol{\Sigma}_i) \tag{1.11}$$

is a zero-mean, Gaussian, IID (independent and identically distributed) residual sequence, which is generally state dependent as indexed by$i$. Because the residual sequence $\mathbf{r}_t(\boldsymbol{\Sigma}_i)$ is IID, and because $\boldsymbol{\mu}_i$ is constant (i.e., not time-varying) given state i, their sum which gives the observation $\mathbf{o}_t$ is thus also IID given the state. Therefore, the HMM discussed above would produce locally or piecewise stationary sequences. Since the temporal locality in question is confined within state occupation of the HMM, we sometimes use the term stationary-state HMM to explicitly denote such a property.

One simple way to extend a stationary-state HMM so that the observation sequence is no longer state-conditioned IID is as follows. We can modify the constant term $\boldsymbol{\mu}_i$ in Eq. 1.10 to explicitly make it time-varying:

$$\mathbf{o}_t = \mathbf{g}_t(\Lambda_i) + \mathbf{r}_t(\boldsymbol{\Sigma}_i), \tag{1.12}$$

where parametersΛ_i in the deterministic time-trend function$\mathbf{g}_t(\Lambda_i)$ is dependent on state i in the Markov chain. This gives rise to the trended (Gaussian) HMM [23, 24, 62, 32, 45, 69, 122, 18, 50, 87, 128], a special version of a nonstationary-state HMM where the first-order statistics (mean) are time-varying and thus violating a basic condition of wide-sense stationarity.

1.3.2 Simulation of a hidden Markov model

When we view a hidden Markov sequence or an HMM as a model for the information source which has been explicitly depicted in Eq. 1.10, sometimes it is desirable to use this model to generate its samples. This is the problem of simulating the HMM given appropriate values for all model parameters: $\{A, \pi, B\}$ for a discrete HMM or $\{A, \pi, \Lambda\}$ for a continuous-density HMM. The result of the simulation is to produce an observation sequence, $\mathbf{o}_1^T = \mathbf{o}_1, \mathbf{o}_2, \cdots, \mathbf{o}_T$, which obeys the statistical law embedded in the HMM. A simulation process is described in Algorithm 1.1.

1.3.3 Likelihood evaluation of a hidden Markov model

Likelihood evaluation is a basic task needed for speech processing applications involving an HMM that uses a hidden Markov sequence to approximate vectorized speech features.

Algorithm 1.1 Draw Samples from an HMM.

1: **procedure** DRAWFROMHMM(A, π, $P(\mathbf{o}_t|s^{(i)})$)
▷ A is the transition probability
▷ π is the initial state occupation probability
▷ $P(\mathbf{o}_t|s^{(i)})$ is the observation probability given a state (either Eq. 1.7 if discrete or 1.8 if continuous
2: Select an initial state $q_1 = s^{(i)}$ by drawing from the discrete distribution π
3: **for** $t \leftarrow 1; t \leq T; t \leftarrow t+1$ **do**
4: Draw an observation $\mathbf{o}_t$ based on $P(\mathbf{o}_t|s^{(i)})$
5: Make a Markov-chain transition from the current state $q_t = s^{(i)}$ to a new state $q_{t+1} = s^{(j)}$ according to the transition probability a_{ij}, and assign $i \leftarrow j$.
6: **end for**
7: **end procedure**

Let $\mathbf{q}_1^T = (q_1, \ldots, q_T)$ be a finite-length sequence of states in a Gaussian-mixture HMM or GMM-HMM, and let $P(\mathbf{o}_1^T, \mathbf{q}_1^T)$ be the joint likelihood of the observation sequence $\mathbf{o}_1^T = (\mathbf{o}_1, \ldots, \mathbf{o}_T)$ and the state sequence $\mathbf{q}_1^T$. Let $P(\mathbf{o}_1^T|\mathbf{q}_1^T)$ denote the likelihood that the observation sequence $\mathbf{o}_1^T$ is generated by the model conditioned on the state sequence $\mathbf{q}_1^T$.

In the Gaussian-mixture HMM, the conditional likelihood $P(\mathbf{o}_1^T|\mathbf{q}_1^T)$ is in the form of

$$P(\mathbf{o}_1^T|\mathbf{q}_1^T) = \prod_{t=1}^{T} b_i(\mathbf{o}_t) = \prod_{t=1}^{T} \sum_{m=1}^{M} \frac{c_{i,m}}{(2\pi)^{D/2}|\boldsymbol{\Sigma}_{i,m}|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{o}_t - \boldsymbol{\mu}_{i,m})^{\mathrm{T}} \boldsymbol{\Sigma}_{i,m}^{-1}(\mathbf{o}_t - \boldsymbol{\mu}_{i,m})\right] \tag{1.13}$$

On the other hand, the probability of state sequence $\mathbf{q}_1^T$ is just the product of transition probabilities, i.e.,

$$P(\mathbf{q}_1^T) = \pi_{q_1} \prod_{t=1}^{T-1} a_{q_t q_{t+1}}. \tag{1.14}$$

In the remaining of the chapter, for notational simplicity, we consider the case where the initial state distribution has probability of one in the starting state: $\pi_{q_1} = 1$.

Note that the joint likelihood $P(\mathbf{o}_1^T, \mathbf{q}_1^T)$ can be obtained by the product of likelihoods in Eqs.1.13 and 1.14:

$$P(\mathbf{o}_1^T, \mathbf{q}_1^T) = P(\mathbf{o}_1^T|\mathbf{q}_1^T)P(\mathbf{q}_1^T). \tag{1.15}$$

In principle, the total likelihood for the observation sequence can be computed by summing the joint likelihoods in Eq.1.15 over all possible state sequences $\mathbf{q}_1^T$:

$$P(\mathbf{o}_1^T) = \sum_{\mathbf{q}_1^T} P(\mathbf{o}_1^T, \mathbf{q}_1^T). \tag{1.16}$$

However, the amount of this computation is exponential in the length of the observation sequence,T, and hence the naive computation of $P(\mathbf{o}_1^T)$ is not tractable. In the next section we will describe the forward-backward algorithm [6] which computes $P(\mathbf{o}_1^T)$ for the HMM with complexity linear in T.

1.3.4 An algorithm for efficient likelihood evaluation

To describe the algorithm, we first define the forward probabilities by

$$\alpha_t(i) = P(q_t = i, \mathbf{o}_1^t), \quad t = 1, \ldots, T, \tag{1.17}$$

and the backward probabilities by

$$\beta_t(i) = P(\mathbf{o}_{t+1}^T | q_t = i), \quad t = 1, \ldots, T-1, \tag{1.18}$$

both for each state$i$ in the Markov chain. The forward and backward probabilities can be calculated recursively from

$$\alpha_t(j) = \sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij} b_j(\mathbf{o}_t), \quad t = 2, 3, ..., T; \qquad j = 1, 2, ..., N \tag{1.19}$$

$$\beta_t(i) = \sum_{j=1}^{N} \beta_{t+1}(j) a_{ij} b_j(\mathbf{o}_{t+1}), \quad t = T-1, T-2, ..., 1; \qquad i = 1, 2, ..., N \tag{1.20}$$

Proofs of these recursions are given immediately after this subsection. The starting value for the α recursion is, according to the definition in Eq.1.17,

$$\alpha_1(i) = P(q_1 = i, \mathbf{o}_1) = P(q_1 = i) P(\mathbf{o}_1 | q_1) = \pi_i b_i(\mathbf{o}_1), \qquad i = 1, 2, ...N \tag{1.21}$$

and that for the β recursion is chosen as

$$\beta_T(i) = 1, \qquad i = 1, 2, ...N, \tag{1.22}$$

so as to provide the correct values for β_{T-1}according to the definition in Eq.1.18.

To compute the total likelihood $P(\mathbf{o}_1^T)$ in Eq.1.16, we first compute

$$\begin{aligned} P(q_t = i, \mathbf{o}_1^T) &= P(q_t = i, \mathbf{o}_1^t, \mathbf{o}_{t+1}^T) \\ &= P(q_t = i, \mathbf{o}_1^t) P(\mathbf{o}_{t+1}^T | \mathbf{o}_1^t, q_t = i) \\ &= P(q_t = i, \mathbf{o}_1^t) P(\mathbf{o}_{t+1}^T | q_t = i) \end{aligned}$$

$$= \alpha_t(i)\beta_t(i), \tag{1.23}$$

for each state i and $t = 1, 2, ..., T$ using definitions in Eqs.1.17 and 1.18. Note that $P(\mathbf{o}_{t+1}^T|\mathbf{o}_1^t, q_t = i) = P(\mathbf{o}_{t+1}^T|q_t = i)$ because the observations are IID given the state in the HMM. Given this, $P(\mathbf{o}_1^T)$ can be computed as

$$P(\mathbf{o}_1^T) = \sum_{i=1}^{N} P(q_t = i, \mathbf{o}_1^T) = \sum_{i=1}^{N} \alpha_t(i)\beta_t(i). \tag{1.24}$$

Taking $t = T$ in Eq.1.24 and using Eq.1.22 lead to

$$P(\mathbf{o}_1^T) = \sum_{i=1}^{N} \alpha_T(i). \tag{1.25}$$

Thus, strictly speaking, theβ recursion is not necessary for the forward scoring computation, and hence the algorithm is often called the forward algorithm. However, theβ computation is a necessary step for solving the model parameter estimation problem, which will be covered in the next section.

1.3.5 Proofs of the forward and backward recursions

Proofs of the recursion formulas, Eqs.1.19 and 1.20, are provided here, using the total probability theorem, Bayes rule, and using the Markov property and conditional independence property of the HMM.

For the forward probability recursion, we have

$$\begin{aligned}
\alpha_t(j) &= P(q_t = j, \mathbf{o}_1^t) \\
&= \sum_{i=1}^{N} P(q_{t-1} = i,\ q_t = j, \mathbf{o}_1^{t-1}, \mathbf{o}_t) \\
&= \sum_{i=1}^{N} P(q_t = j, \mathbf{o}_t|q_{t-1} = i, \mathbf{o}_1^{t-1}) P(q_{t-1} = i, \mathbf{o}_1^{t-1}) \\
&= \sum_{i=1}^{N} P(q_t = j, \mathbf{o}_t|q_{t-1} = i) \alpha_{t-1}(i) \\
&= \sum_{i=1}^{N} P(\mathbf{o}_t|q_t = j,\ q_{t-1} = i) P(q_t = j|q_{t-1} = i) \alpha_{t-1}(i) \\
&= \sum_{i=1}^{N} b_j(\mathbf{o}_t) a_{ij} \alpha_{t-1}(i).
\end{aligned} \tag{1.26}$$

For the backward probability recursion, we have

$$
\begin{aligned}
\beta_t(i) &= P(\mathbf{o}_{t+1}^T | q_t = i) \\
&= \frac{P(\mathbf{o}_{t+1}^T, q_t = i)}{P(q_t = i)} \\
&= \frac{\sum_{j=1}^N P(\mathbf{o}_{t+1}^T, q_t = i,\ q_{t+1} = j)}{P(q_t = i)} \\
&= \frac{\sum_{j=1}^N P(\mathbf{o}_{t+1}^T | q_t = i, q_{t+1} = j) P(q_t = i, q_{t+1} = j)}{P(q_t = i)} \\
&= \sum_{j=1}^N P(\mathbf{o}_{t+1}^T | q_{t+1} = j) \frac{P(q_t = i,\ q_{t+1} = j)}{P(q_t = i)} \\
&= \sum_{j=1}^N P(\mathbf{o}_{t+2}^T, \mathbf{o}_{t+1} | q_{t+1} = j) a_{ij} \\
&= \sum_{j=1}^N P(\mathbf{o}_{t+2}^T | q_{t+1} = j) P(\mathbf{o}_{t+1} | q_{t+1} = j) a_{ij} \\
&= \sum_{j=1}^N \beta_{t+1}(j) b_j(\mathbf{o}_{t+1}) a_{ij}. \qquad (1.27)
\end{aligned}
$$

1.4 EM Algorithm and Its Application to Learning HMM Parameters

1.4.1 Introduction to EM Algorithm

Despite many unrealistic aspects of the HMM as a model for speech feature sequences, one most important reason for its wide-spread use in speech recognition is the Baum-Welch algorithm developed in 1960's [6], which is a prominent instance of the highly popular EM (Expectation-Maximization) algorithm [22], for efficient training of the HMM parameters from data. In this section, we describe first the general principle of the EM algorithm. Then we move to its application to the HMM parameter estimation problem, where the special method of EM becomes known as the Baum-Welch algorithm. For tutorial material on the EM and its basic applications, see [10, 12, 96, 66, 42] .

When there are hidden or latent random variables in a statistical model, maximum likelihood estimation is often difficult and the EM algorithm often becomes effective. Let's denote the complete data by $\mathbf{y} = \{\mathbf{o}, \mathbf{h}\}$ where $\mathbf{o}$ is partially observed data (e.g., speech feature sequence data) and $\mathbf{h}$ is hidden random variables (e.g., unobserved HMM state sequence). Here we consider the problem of finding an estimate for the unknown parameter θ which re-

quires maximization of the log-likelihood function, $\log p(\mathbf{o};\theta)$. However, we may find that this is either too difficult or there are difficulties in finding an expression for the PDF itself. In such circumstances, an iterative solution is possible if the *complete* data,$\mathbf{y}$, can be found such that the PDF in terms of $\mathbf{y}$ is much easier to express in closed form and to maximize. In general we can find a mapping from the complete to the incomplete or partial data: $\mathbf{o} = \mathbf{g}(\mathbf{y})$. However this is usually not evident until one is able to define what the complete data set is. Unfortunately defining what constitutes the complete data is usually an arbitrary procedure that is highly problem specific and often requires some ingenuity on the part of the algorithm designer.

As a way to motivate the EM algorithm, we wish to overcome the computational difficulty of direct optimization of the PDF on the partially observed data $\mathbf{o}$. To accomplish this, we supplement the available data $\mathbf{o}$ with *imaginary* missing, unobserved, or hidden data $\mathbf{h}$, to form the complete data$\mathbf{y}$. The hope is that with a clever choice of the hidden data $\mathbf{h}$, we can work on the complete data$\mathbf{y}$ rather than on the original partial data$\mathbf{o}$ to make the optimization easier for the log likelihood of $\mathbf{o}$.

Once we have identified the complete data $\mathbf{y}$, even though an expression for$\log p(\mathbf{y};\theta)$ can now be derived easily, we cannot directly maximize$\log p(\mathbf{y};\theta)$ with respect to θ since $\mathbf{y}$ is unavailable. However, we observed $\mathbf{o}$and if we further assume that we have a good *guessed* estimate for θ then we can consider the expected value of $\log p(\mathbf{y};\theta)$ conditioned on what we have observed, or the following conditional expectation:

$$Q(\theta|\theta_0) = E_{h|o}[\log p(\mathbf{y};\theta)|\mathbf{o};\theta_0] = E[\log p(\mathbf{o},\mathbf{h};\theta)|\mathbf{o};\theta_0] \tag{1.28}$$

and we attempt to maximize this expectation to yield, not the maximum likelihood estimate, but the next *best* estimate for θ given the previously available estimate of θ_0.

Using Eq.1.28 for computing the conditional expectation when hidden vector$\mathbf{h}$ is continuous, we have

$$Q(\theta|\theta_0) = \int p(\mathbf{h}|\mathbf{o};\theta_0) \log p(\mathbf{y};\theta) d\mathbf{h}. \tag{1.29}$$

When the hidden vector $\mathbf{h}$ is discrete (i.e., taking only discrete values), Eq.1.28 is used to evaluate the conditional expectation:

$$Q(\theta|\theta_0) = \sum_{\mathbf{h}} P(\mathbf{h}|\mathbf{o};\theta_0) \log p(\mathbf{y};\theta) \tag{1.30}$$

where$P(\mathbf{h}|\mathbf{o};\theta_0)$ is a conditional distribution given the initial parameter estimate θ_0, and the summation is over all possible discrete-valued vectors that$\mathbf{h}$ may take.

Given the initial parameters θ_0, the EM algorithm iterates alternating between the E-step, which finds an appropriate expression for the condi-

tional expectation and sufficient statistics for its computation, and the M-step, which maximizes the conditional expectation, until either the algorithm converges or other stopping criteria are met.

Convergence of the EM algorithm is guaranteed (under mild conditions) in the sense that the average log-likelihood of the complete data does not decrease at each iteration, that is

$$Q(\theta|\theta_{k+1}) \geq Q(\theta|\theta_k)$$

with equality when $\theta_k$is already an maximum-likelihood estimate.

The main properties of the EM algorithm are:

- It gives only a local, rather than the global, optimum in the likelihood of partially observed data.
- An initial value for the unknown parameter is needed, and as with most iterative procedures a good initial estimate is required for desirable convergence and a good maximum-likelihood estimate.
- The selection of the complete data set is arbitrary.
- Even if $\log p(\mathbf{y};\theta)$can usually be easily expressed in closed form, finding the closed-form expression for the expectation is usually hard.

1.4.2 Applying EM to learning the HMM — Baum-Welch algorithm

We now discuss how maximal-likelihood parameter estimation and, in particular, the EM algorithm is applied to solve the learning problem for the HMM. As introduced in the preceding section, the EM algorithm is a general iterative technique for maximum likelihood estimation, with local optimality in general, when hidden variables exist. When such hidden variables take the form of a Markov chain, the EM algorithm becomes the Baum-Welch algorithm. Below we use a Gaussian HMM as the example to describe steps involved in deriving E-step and M-step computations, where the complete data in the general case of EM above consists of the observation sequence and the hidden Markov-chain state sequence; i.e.,$\boldsymbol{y} = [\mathbf{o}_1^T, \mathbf{q}_1^T]$.

Each iteration in the EM algorithm consists of two steps for any incomplete data problem including the current HMM parameter estimation problem. In the E (expectation) step of the Baum-Welch algorithm, the following conditional expectation, or the auxiliary function $Q(\theta|\theta_0)$, need to be computed:

$$Q(\theta|\theta_0) = E[\log P(\mathbf{o}_1^T, \mathbf{q}_1^T|\theta)|\mathbf{o}_1^T, \theta_0], \tag{1.31}$$

where the expectation is taken over the "hidden" state sequence $\mathbf{q}_1^T$. For the EM algorithm to be of utility, $Q(\theta|\theta_0)$ has to be sufficiently simplified so that the M (maximization) step can be carried out easily. Estimates of the model

parameters are obtained in the M step via maximization of $Q(\theta|\theta_0)$, which is in general much simpler than direct procedures for maximizing $P(\mathbf{o}_1^T|\theta)$.

An iteration of the above two steps will lead to maximum likelihood estimates of model parameters with respect to the objective function $P(\mathbf{o}_1^T|\theta)$. This is a direct consequence of Baum's inequality [6], which asserts that

$$\log\left(\frac{P(\mathbf{o}_1^T|\theta)}{P(\mathbf{o}_1^T|\theta_0)}\right) \geq Q(\theta|\theta_0) - Q(\theta_0|\theta_0) = 0.$$

We now carry out the E- and M- steps for the Gaussian HMM below, including detailed derivations.

1.4.2.1 E-step

The goal of the E-step is to simplify the conditional expectation $Q(\theta|\theta_0)$ into a form suitable for direct maximization in the M-step. To proceed, we first explicitly write out the $Q(\theta|\theta_0)$ function in terms of expectation over state sequences $\mathbf{q}_1^T$ in the form of a weighted sum

$$Q(\theta|\theta_0) = E[\log P(\mathbf{o}_1^T, \mathbf{q}_1^T|\theta)|\mathbf{o}_1^T, \theta_0] = \sum_{\mathbf{q}_1^T} P(\mathbf{q}_1^T|\mathbf{o}_1^T, \theta_0) \log P(\mathbf{o}_1^T, \mathbf{q}_1^T|\theta), \tag{1.32}$$

where θ and θ_0 denote the HMM parameters in the current and the immediately previous EM iterations, respectively. To simplify the writing, denote by $N_t(i)$ the quantity

$$-\frac{D}{2}\log(2\pi) - \frac{1}{2}\log|\boldsymbol{\Sigma}_i| - \frac{1}{2}(\mathbf{o}_t - \boldsymbol{\mu}_i)^T\boldsymbol{\Sigma}_i^{-1}(\mathbf{o}_t - \boldsymbol{\mu}_i).$$

which is logarithm of the Gaussian PDF associated with state i.

We now use $P(\mathbf{q}_1^T) = \prod_{t=1}^{T-1} a_{q_t q_{t+1}}$ and $P(\mathbf{o}_1^T, \mathbf{q}_1^T) = P(\mathbf{o}_1^T|\mathbf{q}_1^T)P(\mathbf{q}_1^T)$. These lead to

$$\log P(\mathbf{o}_1^T, \mathbf{q}_1^T|\theta) = \sum_{t=1}^{T} N_t(q_t) + \sum_{t=1}^{T-1} \log a_{q_t q_{t+1}}$$

and the conditional expectation in Eq. 1.32 can be rewritten as

$$Q(\theta|\theta_0) = \sum_{\mathbf{q}_1^T} P(\mathbf{q}_1^T|\mathbf{o}_1^T, \theta_0) \sum_{t=1}^{T} N_t(q_t) + \sum_{\mathbf{q}_1^T} P(\mathbf{q}_1^T|\mathbf{o}_1^T, \theta_0) \sum_{t=1}^{T-1} \log a_{q_t q_{t+1}}. \tag{1.33}$$

To simplify $Q(\theta|\theta_0)$ here, we write the first term in Eq.1.33 as

$$Q_1(\theta|\theta_0) = \sum_{i=1}^{N}\{\sum_{\mathbf{q}_1^T} P(\mathbf{q}_1^T|\mathbf{o}_1^T,\theta_0)\sum_{t=1}^{T} N_t(q_t)\}\delta_{q_t,i}, \tag{1.34}$$

and the second term as

$$Q_2(\theta|\theta_0) = \sum_{i=1}^{N}\sum_{j=1}^{N}\{\sum_{\mathbf{q}_1^T} P(\mathbf{q}_1^T|\mathbf{o}_1^T,\theta_0)\sum_{t=1}^{T-1}\log\, a_{q_t q_{t+1}}\}\delta_{q_t,i}\delta_{q_{t+1},j}, \tag{1.35}$$

where δ indicates the Kronecker delta function. Let's examine Eq.1.34 first. By exchanging summations and using the obvious fact that

$$\sum_{\mathbf{q}_1^T} P(\mathbf{q}_1^T|\mathbf{o}_1^T,\theta_0)\delta_{q_t,i} = P(q_t = i|\mathbf{o}_1^T,\theta_0),$$

we can simplify Q_1 into

$$Q_1(\theta|\theta_0) = \sum_{i=1}^{N}\sum_{t=1}^{T} P(q_t = i|\mathbf{o}_1^T,\theta_0)N_t(i). \tag{1.36}$$

After carrying out similar steps for $Q_2(\theta|\theta_0)$ in Eq.1.35 we obtain a similar simplification

$$Q_2(\theta|\theta_0) = \sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{t=1}^{T-1} P(q_t = i,\ q_{t+1} = j|\mathbf{o}_1^T,\theta_0)\log a_{ij}. \tag{1.37}$$

We note that in maximizing $Q(\theta|\theta_0) = Q_1(\theta|\theta_0) + Q_2(\theta|\theta_0)$, the two terms can be maximized independently. That is, $Q_1(\theta|\theta_0)$ contains only the parameters in Gaussians, while $Q_2(\theta|\theta_0)$ involves just the parameters in the Markov chain. Also, in maximizing $Q(\theta|\theta_0)$, the weights in Eq.1.36 and Eq.1.37, or $\gamma_t(i) = P(q_t = i|\mathbf{o}_1^T,\theta_0)$ and $\xi_t(i,j) = P(q_t = i, q_{t+1} = j|\mathbf{o}_1^T,\theta_0)$, respectively, are treated as known constants due to their conditioning on θ_0. They can be computed efficiently via the use of the forward and backward probabilities discussed earlier.

The posterior state transition probabilities in the Gaussian HMM are

$$\xi_t(i,j) = \frac{\alpha_t(i)\beta_{t+1}(j)a_{ij}\exp(N_{t+1}(j))}{P(\mathbf{o}_1^T|\theta_0)}, \tag{1.38}$$

for $t = 1, 2, ..., T-1$. (Note that $\xi_T(i,j)$ has no definition.) The posterior state occupancy probabilities can be obtained by summing $\xi_t(i,j)$ over all the destination states j according to

$$\gamma_t(i) = \sum_{j=1}^{N}\xi_t(i,j), \tag{1.39}$$

for $t = 1, 2, ..., T-1$. $\gamma_T(i)$ can be obtained by its very definition:

$$\gamma_T(i) = P(q_T = i|\mathbf{o}_1^T, \theta_0) = \frac{P(q_T = i, \mathbf{o}_1^T|\theta_0)}{P(\mathbf{o}_1^T|\theta_0)} = \frac{\alpha_T(i)}{P(\mathbf{o}_1^T|\theta_0)}. \tag{1.40}$$

Note for the left-to-right HMM, $\gamma_T(i)$ has a value of one for $i = N$ and of zero otherwise.

Further, we note that the summations in Eqs.1.36 and 1.37 are taken over states i or over state pairs i, j, which is significantly simpler than the summations over state sequences $\mathbf{q}_1^T$ as in the unsimplified forms of $Q_1(\theta|\theta_0)$ and $Q_2(\theta|\theta_0)$ in Eq.1.33. Eqs.1.36 and 1.37 are the simplistic form of the auxiliary objective function, which can be maximized in the M-step discussed next.

1.4.2.2 M-step

The re-estimation formulas for the transition probabilities of the Markov chain in the Gaussian HMM can be easily established by setting $\frac{\partial Q_2}{\partial a_{ij}} = 0$, for Q_2 in Eq.1.37 and for $i, j = 1, 2, ..., N$, subject to the constraint $\sum_{j=1}^{N} a_{ij} = 1$. The standard Lagrange multiplier procedure leads to the re-estimation formula of

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)}, \tag{1.41}$$

where $\xi_t(i, j)$ and $\gamma_t(i)$ are computed according to Eqs.1.38 and 1.39.

To derive the re-estimation formulas for the parameters in the state-dependent Gaussian distributions, we first remove optimization-independent terms and factors in Q_1 in Eq.1.36. Then we have an equivalent objective function of

$$Q_1(\boldsymbol{\mu_i}, \boldsymbol{\Sigma}_i) = \sum_{i=1}^{N} \sum_{t=1}^{\mathrm{Tr}} \gamma_t(i) \left(\mathbf{o}_t - \boldsymbol{\mu}_i\right)^{\mathsf{T}} \boldsymbol{\Sigma}_i^{-1} \left(\mathbf{o}_t - \boldsymbol{\mu}_i\right) - \frac{1}{2} \log |\boldsymbol{\Sigma}_i|. \tag{1.42}$$

The re-estimation formula for the covariance matrices are obtained by solving

$$\frac{\partial Q_1}{\partial \boldsymbol{\Sigma}_i} = 0, \tag{1.43}$$

for $i = 1, 2, ..., N$.

For solving it, we employ the trick of variable transformation: $\mathbf{K} = \boldsymbol{\Sigma}^{-1}$ (we omit the state index i for simplicity), and we treat Q_1 as a function of $\mathbf{K}$. Then, the derivative of $\log |\mathbf{K}|$ (a term in Eq.1.36) with respect to $\mathbf{K}$'s lm-th entry,k_{lm}, is the lm-th entry of Σ, orσ_{lm}. We now can reduce $\frac{\partial Q_1}{\partial k_{lm}} = 0$ to

$$\sum_{t=1}^{T} \gamma_t(i)\{\frac{1}{2}\sigma_{lm} - \frac{1}{2}(\mathbf{o}_t - \boldsymbol{\mu}_i)_l(\mathbf{o}_t - \boldsymbol{\mu}_i))_m\} = 0 \tag{1.44}$$

for each entry: $l, m = 1, 2, ..., D$. Writing this result in a matrix form, we obtain the compact re-estimation formula for the covariance matrix in state i as follows:

$$\hat{\boldsymbol{\Sigma}}_i = \frac{\sum_{t=1}^{T} \gamma_t(i)(\mathbf{o}_t - \hat{\boldsymbol{\mu}}_i)(\mathbf{o}_t - \hat{\boldsymbol{\mu}}_i)^{\mathsf{T}}}{\sum_{t=1}^{T} \gamma_t(i)} \tag{1.45}$$

for each state: $i = 1, 2, ..., N$, where $\hat{\boldsymbol{\mu}}_i$is the re-estimate of the mean vectors in the Gaussian HMM in state i, whose re-estimation formula is straightforward to derive and has the following easily interpretable form:

$$\hat{\boldsymbol{\mu}}_i = \frac{\sum_{t=1}^{T} \gamma_t(i)\mathbf{o}_t}{\sum_{t=1}^{T} \gamma_t(i)} \tag{1.46}$$

The above derivation is for the single-Gaussian HMM. The EM algorithm for the GMM-HMM can be similarly derived by considering the Gaussian component of each frame at each state as another hidden variable. In Chapter **??** we will describe the deep neural network (DNN)-HMM hybrid system in which the observation probability is estimated using a DNN.

1.5 Viterbi Algorithm for Decoding HMM State Sequences

1.5.1 Dynamic programming and Viterbi algorithm

Dynamic Programming (DP) is a divide-and-conquer method for solving complex problems by breaking them down into simpler subproblemsis [7, 110]. It was originally developed by R. Bellman in 1950s [7]. The foundation of DP was laid by the Bellman optimality principle. The principle stipulates that: "In an optimization problem concerning multiple stages of interrelated decisions, whatever the initial state (or condition) and the initial decisions are, the remaining decisions must constitute an optimal rule of choosing decisions with regards to the state that results from the first decision."

As an example, we discuss the optimality principle in Markov decision processes. A Markov decision process is characterized by two sets of parameters. The first set is the transition probabilities of

$$P_{ij}^{k}(n) = P(\text{state}_\text{j}, \text{stage}_{\text{n}+1} | \text{state}_\text{i}, \text{stage}_\text{n}, \text{decision}_\text{k}),$$

where the current state of the system is dependent only on the state of the system at the previous stage and the decision taken at that stage (Markov property). The second set of parameters provide rewards defined by:

$R_i^k(n)$ =reward at stage n and at state i when decision k is chosen.

Let us define $F(n,i)$ to be the average of the total reward at state n and state i when the optimal decision is taken. This can be computed by DP using the following recursion given by the optimality principle:

$$F(n,i) = \max_k \{R_i^k(n) + \sum_j P_{ij}^k(n) F(n+1,j)\}. \tag{1.47}$$

In particular, when $n = N$ (at the final stage), the total reward at state i is

$$F(N,i) = \max_k R_i^k(N). \tag{1.48}$$

The optimal decision sequence can be traced back after the end of this recursive computation.

We see in the above that in applying DP, various stages (e.g., stage $1, 2, ..., n, ...N$ in the above example) in the optimization process must be identified. We are required at each stage to make optimal decision(s). There are several states (indexed by i in the above example) of the system associated with each stage. The decision (indexed by k) taken at a given stage changes the problem from the current stage n to the next stage $n+1$ according to the transition probability $P_{ij}^k(n)$.

If we apply the DP technique to finding the optimal path, then the Bellman optimality principle can be alternatively stated as follows: "The optimal path from nodes A to C through node B must consist of the optimal path from A to B concatenated with the optimal path from B to C." The implication of this optimality principle is tremendous. That is, in order to find the best path from node A via a "predecessor" node B, there will be no need to reconsider all the partial paths leading from A to B. This significantly reduces the path search effort compared with the brute-force search or exhaustive search. While it may be unknown whether the "predecessor" node B is on the best path or not, many candidates can be evaluated and the correct one be ultimately determined via a backtracking procedure in DP. The Bellman optimality principle is the essence of a very popular optimization technique in speech processing applications involving the HMM, which we describe below.

1.5.2 Dynamic programming for decoding HMM states

One fundamental computational problem associated with the HMM discussed so far in this chapter is to find, in an efficient manner, the best sequence of the HMM states given an arbitrary sequence of observations $\mathbf{o}_1^T = \mathbf{o}_1, \mathbf{o}_2, \cdots, \mathbf{o}_T$.

This is a complex T-stage path-finding optimization problem, and is directly suited for the DP solution. The DP technique used for such purposes is also called the Viterbi algorithm, developed originally for optimal convolution-code channel decoding in digital communication.

To illustrate the Viterbi algorithm as an optimal path-finding technique, we can use the two-dimensional grid, also called the trellis diagram, for a left-to-right HMM. A node in the trellis diagram is associated with both a time frame$t$ on the horizontal axis and an HMM state i on the vertical axis.

For a given HMM characterized by the state transitional probabilities a_{ij} and by the state-conditioned output probability distributions $b_i(\mathbf{o}_t)$, let $\delta_i(t)$ represent the maximal value of the joint likelihood of the partial observation sequence $\mathbf{o}_1^t$ up to time t, and the associated HMM state sequence while in state i at time t. That is,

$$\delta_i(t) = \max_{q_1, q_2, \ldots, q_{t-1}} P(\mathbf{o}_1^t,\, q_1^{t-1},\, q_t = i). \tag{1.49}$$

Note that each $\delta_i(t)$ defined here is associated with a node in the trellis diagram. Each increment of time corresponds to reaching a new stage in DP. At the final stage $t = T$, we have the objective function of $\delta_i(T)$ which is accomplished via all the previous stages of computation for$t \leq T - 1$. Based on the DP optimality principle, the optimal partial likelihood of Eq. 1.50 at the processing stage of $t + 1$ can be computed using the following functional equation as a recursion:

$$\delta_j(t+1) = \max_i \delta_i(t) a_{ij} b_j(\mathbf{o}_{t+1}), \tag{1.50}$$

for each state j. Each state at this processing stage is a hypothesized "precursor" node in the global optimal path. All such nodes except one will be eventually eliminated after the backtracking operation. The essence of DP used here is that we only need to compute the quantities of $\delta_j(t + 1)$ as individual nodes in the trellis, removing the need to keep track of a very large number of partial paths from the initial stage to the current $(t + 1)^{th}$ stage, which would be required for the exhaustive search. The optimality is guaranteed, due to the DP optimality principle, with the computation only linearly, rather than geometrically, increasing with the length T of the observation data sequence.

Besides the key recursion of Eq.1.50, the complete Viterbi algorithm requires additional steps of recursion initialization, recursion termination, and path backtracking. The complete algorithm is described in Algorithm 1.2 with initial state probabilities π_i. The result of the Viterbi algorithm is P^*, the maximum joint likelihood of the observation and state sequence, together with $q^*(t)$, the corresponding state transition path.

The optimal state transition path found by the above Viterbi algorithm for a left-to-right HMM is equivalent to the information required to determine the optimal segmentation of the HMM states. The concept of state segmentation

Algorithm 1.2 Viterbi algorithm for decoding HMM state sequence.

procedure VITERBIDECODE($A = [a_{ij}]$, π, $b_j(\mathbf{o}_t)$)
▷ A is the transition probability
▷ π is the initial state occupation probability
▷ $b_j(\mathbf{o}_t)$ is observation probability given HMM state j for observation vector $\mathbf{o}_t$
  $\delta_i(1) \leftarrow \pi_i b_i(\mathbf{o}_1)$ ▷ Initialize at $t = 1$
  $\psi_i(1) \leftarrow 0$ ▷ Initialize at $t = 1$
  for $t \leftarrow 2; t \leq T; t \leftarrow t + 1$ **do**
    ▷ Forward recursion
    $\delta_j(t) \leftarrow \max_i \delta_i(t-1) a_{ij} b_j(\mathbf{o}_t)$
    $\psi_j(t) \leftarrow \arg\ \max_{1 \leq i \leq N} \delta_i(t-1) a_{ij}$
  end for
  $P^* \leftarrow \max_{1 \leq i \leq N} [\delta_i(T)]$
  $q(T) \leftarrow \max_{1 \leq i \leq N} [\delta_i(T)]$ ▷ Initialize backtracking
  for $t \leftarrow T-1; t \geq 1; t \leftarrow t-1$ **do**
    ▷ Backward state tracking
    $q^*(t) \leftarrow \psi_{q^*(t+1)}(t+1)$
  end for
  Return optimal HMM state path $q^*(t)$, $1 \leq t \leq T$
end procedure

is most relevant to a left-to-right HMM commonly used in speech modeling and recognition, as each state in such an HMM is typically associated with a reasonably large number of consecutive time frames in the observation sequence. This is so because the observations cannot be easily assigned back to earlier states due to the left-to-right constraint and because the last frame must be accounted for by the right-most state in the left-to-right HMM.

Note that this same Viterbi algorithm can be applied to single-Gaussian HMM, GMM-HMM and even the DNN-HMM we will describe in Chapter **??**.

1.6 The HMM and Variants for Generative Speech Modeling and Recognition

The popularity of the HMM in speech recognition stems from its ability as a generative sequence model of acoustic features of speech. See excellent reviews of the HMM for selected speech modeling and recognition applications in [106, 105, 107, 72, 3, 4]. One most interesting and unique problem in speech modeling and in the related speech recognition application lies in the nature of variable length in acoustic-feature sequences. This unique characteristic of speech rests primarily in its temporal dimension. That is, the actual values of the speech feature are correlated lawfully with the elasticity in the temporal dimension. As a consequence, even if two word sequences are identical, the acoustic data of speech features typically have distinct lengths. For example,

different acoustic samples from the same sentence usually contain different data dimensionality, depending on how the speech sounds are produced and in particular how fast the speaking rate is. Further, the discriminative cues among separate speech classes are often distributed over a reasonably long temporal span, which often crosses neighboring speech units. Other special aspects of speech include class-dependent acoustic cues. These cues are often expressed over diverse time spans that would benefit from different lengths of analysis windows in speech analysis and feature extraction.

Conventional wisdom posits that speech is a one-dimensional temporal signal in contrast to image and video as higher dimensional signals. This view is simplistic and does not capture the essence and difficulties of the speech recognition problem. Speech is best viewed as a two-dimensional signal, where the spatial (or frequency or tonotopic) and temporal dimensions have vastly different characteristics, in contrast to images where the two spatial dimensions tend to have similar properties. The "spatial" dimension in speech is associated with the frequency distribution and related transformations, capturing a number of variability types including primarily those arising from environments, speakers, accent, speaking style and rate. The latter induces correlations between spatial and temporal dimensions, and the environment factors include microphone characteristics, speech transmission channel, ambient noise, and room reverberation.

The temporal dimension in speech, and in particular its correlation with the spatial or frequency-domain properties of speech, constitutes one of the unique challenges for speech recognition. The HMM addresses this challenge to a limited extent. In this section, a selected set of advanced generative models, as various extensions of the HMM, will be described that are aimed to address the same challenge, where Bayesian approaches are used to provide temporal constraints as prior knowledge about aspects of the physical process of human speech production.

1.6.1 GMM-HMMs for speech modeling and recognition

In speech recognition, one most common generative learning approach is based on the Gaussian-Mixture-Model based Hidden Markov models, or GMM-HMM; e.g., [74, 105, 39, 107, 12]. As discussed earlier, a GMM-HMM is a statistical model that describes two dependent random processes, an observable process and a hidden Markov process. The observation sequence is assumed to be *generated* by each hidden state according to a Gaussian mixture distribution. A GMM-HMM is parameterized by a vector of state prior probabilities, the state transition probability matrix, and by a set of state-dependent parameters in Gaussian mixture models. In terms of modeling speech, a state in the GMM-HMM is typically associated with a subsegment of a phone in speech. One important innovation in the use of HMMs

for speech recognition is the introduction of context-dependent states (e.g., [40, 70]), motivated by the desire to reduce output variability of speech feature vectors associated with each state, a common strategy for "detailed" generative modeling. A consequence of using context dependency is a vast expansion of the HMM state space, which, fortunately, can be controlled by regularization methods such as state tying. It turns out that such context dependency also plays a critical role in the recent advance of speech recognition in the area of discrimination-based deep learning [121, 19, 20, 112, 16], to be discussed in later chapters of this book.

The introduction of the HMM and the related statistical methods to speech recognition in mid 1970s [3, 72] can be regarded as the most significant paradigm shift in the field, as discussed and analyzed in [4, 5]. One major reason for this early success is the highly efficient EM algorithm [6], which we described earlier in this chapter. This maximum likelihood method, often called Baum-Welch algorithm, had been a principal way of training the HMM-based speech recognition systems until 2002, and is still one major step (among many) in training these systems nowadays. It is interesting to note that Baum-Welch algorithm serves as one major motivating example for the later development of the more general EM algorithm [22]. The goal of maximum likelihood or EM method in training GMM-HMM speech recognizers is to minimize the empirical risk with respect to the joint likelihood loss involving a sequence of linguistic labels and a sequence of acoustic data of speech, often extracted at the frame level. In large-vocabulary speech recognition systems, it is normally the case that word-level labels are provided, while state-level labels are latent. Moreover, in training GMM-HMM-based speech recognition systems, parameter tying is often used as a type of regularization. For example, similar acoustic states of the triphones can share the same Gaussian mixture model.

The use of the generative model of HMMs for representing the (piecewise stationary) dynamic speech pattern and the use of EM algorithm for training the tied HMM parameters constitute one most prominent and successful example of generative learning in speech recognition. This success has been firmly established by the speech community, and has been widely spread to machine learning and related communities. In fact, the HMM has become a standard tool not only in speech recognition but also in machine learning as well as their related fields such as bioinformatics and natural language processing. For many machine learning as well as speech recognition researchers, the success of HMMs in speech recognition is a bit surprising due to the well-known weaknesses of the HMM in modeling speech dynamics. The remaining part of this section is aimed to address ways of using more advanced dynamic generative models and related techniques for speech modeling and recognition.

1.6.2 Trajectory and hidden dynamic models for speech modeling and recognition

Despite great success of GMM-HMMs in speech modeling and recognition, their weaknesses, such as the conditional independence and piecewise stationary assumptions, have been well known for speech modeling and recognition applications since early days[23, 98, 24, 32, 97, 46, 15, 51]. Since early 1990's, speech recognition researchers have begun the development of statistical models that capture more realistic dynamic properties of speech in the temporal dimension than HMMs do. This class of extended HMM models have been variably called stochastic segment model [98, 97], trended or nonstationary-state HMM [23, 32, 18], trajectory segmental model [69, 97], trajectory HMM [128, 126], stochastic trajectory model [62], hidden dynamic model [44, 25, 15, 100, 89, 90, 91, 109, 29], buried Markov model [11, 14, 13], structured speech model, and hidden trajectory model [130, 52, 51, 29, 50, 120, 119], depending on different "prior knowledge" applied to the temporal structure of speech and on various simplifying assumptions to facilitate the model implementation. Common to all these beyond-HMM model variants is some temporal dynamic structure built into the models. Based on the nature of such structure, we can classify these models into two main categories. In the first category are the models focusing on temporal correlation structure at the "surface" acoustic level. The second category consists of deep hidden or latent dynamics, where the underlying speech production mechanisms are exploited as a prior to represent the temporal structure that accounts for the visible speech pattern. When the mapping from the hidden dynamic layer to the visible layer is limited to be linear and deterministic, then the generative hidden dynamic models in the second category reduce to the first category.

The temporal span in many of the generative dynamic/trajectory models above is often controlled by a sequence of linguistic labels, which segment the full sentence into multiple regions from left to right; hence segment models.

In general, the trajectory or segmental models with hidden or latent dynamics make use of the switching state space formulation, well studied in the literature; e.g., [61, 28, 54, 80, 94, 108, 53]. These models exploit temporal recursion to define the hidden dynamics, $\mathbf{z}(k)$, which may correspond to articulatory movements during human speech production. Each discrete region or segment, s, of such dynamics is characterized by the s-dependent parameter set $\boldsymbol{\Lambda}_{\mathbf{s}}$, with the "state noise" denoted by $\mathbf{w}_s(k)$. The memory-less nonlinear mapping function is exploited to link the hidden dynamic vector $\mathbf{z}(k)$ to the observed acoustic feature vector $\mathbf{o}(k)$, with the "observation noise" denoted by $\mathbf{v}_s(k)$, and parameterized also by segment-dependent parameters. This pair of "state equation" and "observation equation" below form a general state-space switching nonlinear dynamic system model:

$$\mathbf{z}(k) = \mathbf{q}_k[\mathbf{z}(k-1), \boldsymbol{\Lambda}_s] + \mathbf{w}_s(k-1) \tag{1.51}$$

$$\mathbf{o}(k') = \mathbf{r}_{k'}[\mathbf{z}(k'), \boldsymbol{\Omega}_{s'}] + \mathbf{v}_{s'}(k'), \tag{1.52}$$

where subscripts k and k' denote that functions $\mathbf{q}[.]$ and $\mathbf{r}[.]$ are time varying and may be asynchronous with each other. In the mean time, s or s' denotes the dynamic region that is correlated with discrete linguistic categories either in terms of allophone states as in the standard GMM-HMM system (e.g., [105, 40, 70]) or in terms of atomic units constructed from articulation-motivated phonological features (e.g., [88, 78, 113, 59, 47, 26]).

The speech recognition literature has reported a number of studies on switching nonlinear state space models, both theoretical and experimental. The specific forms of the functions of $\mathbf{q}_k[\mathbf{z}(k-1), \boldsymbol{\Lambda}_s]$ and $\mathbf{r}_{k'}[\mathbf{z}(k'), \boldsymbol{\Omega}_{s'}]$ and their parameterization are determined by prior knowledge based on the understanding of the temporal properties of speech. In particular, state equation equation 1.51 takes into account the temporal elasticity in spontaneous speech and its correlation with the "spatial" properties in hidden speech dynamics such as articulatory positions or vocal tract resonance frequencies. For example, these latent variables do not oscillate within each phone-bound temporal region. And observation equation 1.52 incorporates knowledge about forward, nonlinear mapping from articulation to acoustics, an intensely studied subject in speech production and speech analysis research [77, 34, 35, 33].

When nonlinear functions of $\mathbf{q}_k[\mathbf{z}(k-1), \boldsymbol{\Lambda}_s]$ and $\mathbf{r}_{k'}[\mathbf{z}(k'), \boldsymbol{\Omega}_{s'}]$ are reduced to linear functions (and when synchrony between the two equations are eliminated), the switching nonlinear dynamic system model is reduced to its linear counterpart, or the switching linear dynamic system. This simplified system can be viewed as a hybrid of the standard HMM and linear dynamical systems, one associated with each HMM state. The general mathematical description of the switching linear dynamic system can be written as

$$\mathbf{z}(k) = \mathbf{A}_s\mathbf{z}(k-1) + \mathbf{B}_s\mathbf{w}_s(k) \tag{1.53}$$

$$\mathbf{o}(k) = \mathbf{C}_s\mathbf{z}(k) + \mathbf{v}_s(k). \tag{1.54}$$

where subscript s denotes the left-to-right HMM state or the region of the switching state in the linear dynamics. There has been an interesting set of work on the switching linear dynamic system applied to speech recognition. The early set of studies have been were reported in [98, 97] for generative speech modeling and for speech recognition applications. More recent studies [53, 94] applied linear switching dynamic systems to noise-robust speech recognition and explored several approximate inference techniques. The study reported in [108] applied another approximate inference technique, a special type of Gibbs sampling, to a speech recognition problem.

1.6.3 The speech recognition problem using generative models of HMM and its variants

Toward the end of this chapter, let us focus on a discussion on issues related to using generative models such as the standard HMM and its extended versions just described for discriminative classification problems such as speech recognition. More detailed discussions on this important topic can be found in [41][129, 57]. In particular, we have omitted in this chapter the topic of discriminative learning of the generative model of HMM, very important in the development of ASR based on the GMM-HMM and related architectures. We leave the readers to a plethora of literature on this topic in [2, 73, 17, 18, 111, 9, 103, 104, 116, 92, 93, 102, 123, 124, 127, 114, 125, 65, 63, 101, 67, 64, 56, 48, 66, 117]. Another important topic which we also omitted in this chapter is the use of the generative, GMM-HMM-based models for integrating the modeling of the effects of noise in ASR. The ability to naturally carry out such integrated modeling is one of the strengths of statistical generative models such as the GMM and HMM, which we also leave the readers to the literature including many review articles [86, 58, 1, 55, 31, 48, 37, 36, 38, 49, 83, 94, 84, 75, 85, 60, 76, 115, 30].

A generative statistical model characterizes joint probabilities of input data and their corresponding labels, which can be decomposed into the prior probability of the labels (e.g., speech class tags) and the probability of class-conditioned data (e.g., acoustic features of speech). Via Bayes rule, the posterior probability of class labels given the data can be easily determined and used as the basis for the decision rule for classification. One key factor for the success of such generative modeling approach in classification tasks is how good the model is to the true data distribution. The HMM has been shown to be a reasonably good model for the statistical distribution of sequence data of speech acoustics, especially in its temporal characteristics. As a result, the HMM has become a popular model for speech recognition since mid 1980's.

However, several weaknesses of the standard HMM as a generative model for speech have been well understood, including, among others, the temporal independence of speech data conditioned on each HMM state and lack of lawful correlation between the acoustic features and ways in which speech sounds are produced (e.g., speaking rate and style). These weaknesses have motivated extensions of the HMM in several ways, some discussed in this section. The main thread of these extensions include the replacement of the Gaussian or Gaussian-mixture-like, independent and identical distributions associated with each HMM state by more realistic, temporally correlated dynamic systems or non-stationary trajectory models, both of which contain latent, continuous-valued dynamic structure.

During the development of these hidden trajectory and hidden dynamic models for speech recognition, a number of machine learning techniques, notably approximate variational inference and learning techniques [61, 99, 118,

80], have been usefully applied with modifications and improvement to suit the speech-specific properties and speech recognition applications. However, the success has mostly been limited to relatively small tasks. We can identify four main sources of difficulties (as well as new opportunities) in successfully applying these types of generative models to large-scale speech recognition. First, scientific knowledge on the precise nature of the underlying articulatory speech dynamics and its deeper articulatory control mechanisms is far from complete. Coupled with the need for efficient computation in training and decoding for speech recognition applications, such knowledge was forced to be again simplified, reducing the modeling power and precision further. Second, most of the work in this area has been placed within the generative learning setting, having a goal of providing parsimonious accounts (with small parameter sets) for speech variations due to contextual factors and co-articulation. In contrast, the recent joint development of deep learning methods, which we will cover in several later chapters of this book, combines generative and discriminative learning paradigms and makes use of massive instead of parsimonious parameters. There appears to be a huge potential for synergy of research here, especially in light of the recent progress on variational inference expected to improve the quality of deep, generative modeling and learning [79, 95, 21, 68, 8]. Third, most of the hidden trajectory or hidden dynamic models have focused on only isolated aspects of speech dynamics rooted in deep human production mechanisms, and have been constructed using relatively simple and largely standard forms of dynamic systems without sufficient structure and effective learning methods free from unknown approximation errors especially during the inference step. This latter deficiency can likely be overcome by the improved variational learning methods just discussed.

Functionally speaking, speech recognition is a conversion process from the acoustic data sequence of speech into a word or another linguistic-symbol sequence. Technically, this conversion process requires a number of subprocesses including the use of discrete time stamps, often called frames, to characterize the speech waveform data or acoustic features, and the use of categorical labels (e.g. words, phones, etc.) to index the acoustic data sequence. The fundamental issues in speech recognition lie in the nature of such labels and data. It is important to clearly understand the unique attributes of speech recognition, in terms of both input data and output labels. From the output viewpoint, ASR produces sentences that consist of a variable number of words. Thus, at least in principle, the number of possible classes (sentences) for the classification is so large that it is virtually impossible to construct models for complete sentences without the use of structure. From the input viewpoint, the acoustic data are also a sequence with a variable length, and typically, the length of data input is vastly different from that of label output, giving rise to the special problem of segmentation or alignment that the "static" classification problems in machine learning do not encounter. Combining the input and output viewpoints, we state the fundamental prob-

lem of speech recognition as a structured sequence classification task, where a (relatively long) sequence of acoustic data is used to infer a (relatively short) sequence of the linguistic units such as words. For this type of structured pattern recognition, both the standard HMM and its variants discussed in this chapter have captured some major attributes of the speech problem, especially in the temporal modeling aspect, accounting for their practical success in speech recognition to some degree. However, other key attributes of the problem have been poorly captured by the many types of models discussed in this chapter. Most of the remaining chapters in this book will be devoted to addressing this deficiency.

As a summary of this section, we bridged the HMM as a generative statistical model to practical speech problems including its modeling and classification/recognition. We pointed out the weaknesses of the standard HMM as a generative model for characterizing temporal properties of speech features, motivating its extensions to several variants where the temporal independence of speech data conditioned on each HMM state is replaced by more realistic, temporally correlated dynamic systems with latent structure. The state-space formulation of nonlinear dynamic system models provides an intriguing mechanism to connect to the recurrent neural networks, which we will discuss in great detail later in Chapter **??**.

References

1. Acero, A., Deng, L., Kristjansson, T.T., Zhang, J.: HMM adaptation using vector taylor series for noisy speech recognition. In: Proc. Annual Conference of International Speech Communication Association (INTERSPEECH), pp. 869–872 (2000)
2. Bahl, L., Brown, P., de Souza, P., Mercer, R.: Maximum mutual information estimation of HMM parameters for speech recognition. In: Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 49–52 (1986)
3. Baker, J.: Stochastic modeling for automatic speech recognition. In: D. Reddy (ed.) Speech Recognition. Academic, New York (1976)
4. Baker, J., Deng, L., Glass, J., Khudanpur, S., Lee, C.H., Morgan, N., O'Shgughnessy, D.: Research developments and directions in speech recognition and understanding, part i. IEEE Signal Processing Magazine **26**(3), 75–80 (2009)
5. Baker, J., Deng, L., Glass, J., Khudanpur, S., Lee, C.H., Morgan, N., O'Shgughnessy, D.: Updated minds report on speech recognition and understanding (research developments and directions in speech recognition and understanding, part ii). IEEE Signal Processing Magazine **26**(4), 78–85 (2009)
6. Baum, L., Petrie, T.: Statistical inference for probabilistic functions of finite state Markov chains. Ann. Math. Statist. **37**(6), 1554—1563 (1966)
7. Bellman, R.: Dynamic Programming. Princeton University Press (1957)
8. Bengio, Y.: Estimating or propagating gradients through stochastic neurons. CoRR (2013)
9. Biem, A., Katagiri, S., McDermott, E., Juang, B.H.: An application of discriminative feature extraction to filter-bank-based speech recognition. IEEE Transactions on Speech and Audio Processing (9), 96–110 (2001)
10. Bilmes, J.: A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. Tech. Rep. TR-97-021, ICSI (1997)
11. Bilmes, J.: Buried markov models: A graphical modeling approach to automatic speech recognition. Computer Speech and Language **17**, 213—231 (2003)
12. Bilmes, J.: What HMMs can do. IEICE Trans. Information and Systems **E89-D**(3), 869–891 (2006)
13. Bilmes, J.: Dynamic graphical models. IEEE Signal Processing Magazine **33**, 29–42 (2010)
14. Bilmes, J., Bartels, C.: Graphical model architectures for speech recognition. IEEE Signal Processing Magazine **22**, 89–100 (2005)

15. Bridle, J., Deng, L., Picone, J., Richards, H., Ma, J., Kamm, T., Schuster, M., Pike, S., Reagan, R.: An investigation fo segmental hidden dynamic models of speech coarticulation for automatic speech recognition. Final Report for 1998 Workshop on Langauge Engineering, CLSP, Johns Hopkins (1998)
16. Chen, X., Eversole, A., Li, G., Yu, D., Seide, F.: Pipelined back-propagation for context-dependent deep neural networks. In: Proc. Annual Conference of International Speech Communication Association (INTERSPEECH) (2012)
17. Chengalvarayan, R., Deng, L.: HMM-based speech recognition using state-dependent, discriminatively derived transforms on mel-warped DFT features. IEEE Transactions on Speech and Audio Processing (5), 243–256 (1997)
18. Chengalvarayan, R., Deng, L.: Speech trajectory discrimination using the minimum classification error learning. IEEE Transactions on Speech and Audio Processing (6), 505–515 (1998)
19. Dahl, G., Yu, D., Deng, L., Acero, A.: Large vocabulary continuous speech recognition with context-dependent DBN-HMMs. In: Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP) (2011)
20. Dahl, G., Yu, D., Deng, L., Acero, A.: Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. IEEE Transactions on Audio, Speech and Language Processing **20**(1), 30–42 (2012)
21. Danilo Jimenez Rezende Shakir Mohamed, D.W.: Stochastic backpropagation and approximate inference in deep generative models. In: Proc. International Conference on Machine Learning (ICML) (2014)
22. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum-likelihood from incomplete data via the EM algorithm. J. Royal Statist. Soc. Ser. B. **39** (1977)
23. Deng, L.: A generalized hidden markov model with state-conditioned trend functions of time for the speech signal. Signal Processing **27**(1), 65–78 (1992)
24. Deng, L.: A stochastic model of speech incorporating hierarchical nonstationarity. IEEE Transactions on Acoustics, Speech and Signal Processing **1**(4), 471–475 (1993)
25. Deng, L.: A dynamic, feature-based approach to the interface between phonology and phonetics for speech modeling and recognition. Speech Communication **24**(4), 299–323 (1998)
26. Deng, L.: Articulatory features and associated production models in statistical speech recognition. In: Computational Models of Speech Pattern Processing, pp. 214–224. Springer-Verlag, New York (1999)
27. Deng, L.: Computational models for speech production. In: Computational Models of Speech Pattern Processing, pp. 199–213. Springer-Verlag, New York (1999)
28. Deng, L.: Switching dynamic system models for speech articulation and acoustics. In: Mathematical Foundations of Speech and Language Processing, pp. 115–134. Springer-Verlag, New York (2003)
29. Deng, L.: DYNAMIC SPEECH MODELS — Theory, Algorithm, and Applications. Morgan and Claypool (2006)
30. Deng, L.: Front-End, Back-End, and Hybrid Techniques to Noise-Robust Speech Recognition. Chapter 4 in Book: Robust Speech Recognition of Uncertain Data. Springer Verlag (2011)
31. Deng, L., Acero, A., Plumpe, M., Huang, X.: Large vocabulary speech recognition under adverse acoustic environment. In: Proc. International Conference on Spoken Language Processing (ICSLP), pp. 806–809 (2000)
32. Deng, L., Aksmanovic, M., Sun, D., Wu, J.: Speech recognition using hidden Markov models with polynomial regression functions as non-stationary states. IEEE Transactions on Acoustics, Speech and Signal Processing **2**(4), 101–119 (1994)

33. Deng, L., Attias, H., Lee, L., Acero, A.: Adaptive kalman smoothing for tracking vocal tract resonances using a continuous-valued hidden dynamic model. IEEE Transactions on Audio, Speech and Language Processing **15**, 13–23 (2007)
34. Deng, L., Bazzi, I., Acero, A.: Tracking vocal tract resonances using an analytical nonlinear predictor and a target-guided temporal constraint. In: Proc. Annual Conference of International Speech Communication Association (INTERSPEECH) (2003)
35. Deng, L., Dang, J.: Speech analysis: The production-perception perspective. In: Advances in Chinese Spoken Language Processing. World Scientific Publishing (2007)
36. Deng, L., Droppo, J., A.Acero: Recursive estimation of nonstationary noise using iterative stochastic approximation for robust speech recognition. IEEE Transactions on Speech and Audio Processing **11**, 568–580 (2003)
37. Deng, L., Droppo, J., Acero, A.: A Bayesian approach to speech feature enhancement using the dynamic cepstral prior. In: Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP), vol. 1, pp. I–829 –I–832 (2002)
38. Deng, L., Droppo, J., Acero, A.: Enhancement of log mel power spectra of speech using a phase-sensitive model of the acoustic environment and sequential estimation of the corrupting noise. IEEE Transactions on Speech and Audio Processing **12**(2), 133 – 143 (2004)
39. Deng, L., Kenny, P., Lennig, M., Gupta, V., Seitz, F., Mermelsten, P.: Phonemic hidden markov models with continuous mixture output densities for large vocabulary word recognition. IEEE Transactions on Acoustics, Speech and Signal Processing **39**(7), 1677–1681 (1991)
40. Deng, L., Lennig, M., Seitz, F., Mermelstein, P.: Large vocabulary word recognition using context-dependent allophonic hidden markov models. Computer Speech and Language **4**, 345–357 (1991)
41. Deng, L., Li, X.: Machine learning paradigms in speech recognition: An overview. IEEE Transactions on Audio, Speech and Language Processing **21**(5), 1060–1089 (2013)
42. Deng, L., Mark, J.: Parameter estimation for markov modulated poisson processes via the em algorithm with time discretization. In: Telecommunication Systems (1993)
43. Deng, L., O'Shaughnessy, D.: SPEECH PROCESSING — A Dynamic and Optimization-Oriented Approach. Marcel Dekker Inc, NY (2003)
44. Deng, L., Ramsay, G., Sun, D.: Production models as a structural basis for automatic speech recognition. Speech Communication **33**(2-3), 93–111 (1997)
45. Deng, L., Rathinavelu, C.: A Markov model containing state-conditioned second-order non-stationarity: application to speech recognition. Computer Speech and Language **9**(1), 63–86 (1995)
46. Deng, L., Sameti, H.: Transitional speech units and their representation by regressive Markov states: Applications to speech recognition. IEEE Transactions on speech and audio processing **4**(4), 301–306 (1996)
47. Deng, L., Sun, D.: A statistical approach to automatic speech recognition using the atomic speech units constructed from overlapping articulatory features. Journal Acoustical Society of America **85**, 2702–2719 (1994)
48. Deng, L., Wang, K., Acero, A., Hon, H., Droppo, J., C. Boulis, Y.W., Jacoby, D., Mahajan, M., Chelba, C., Huang, X.: Distributed speech processing in mipad's multimodal user interface. IEEE Transactions on Audio, Speech and Language Processing **20**(9), 2409 –2419 (2012)
49. Deng, L., Wu, J., Droppo, J., Acero, A.: Analysis and comparisons of two speech feature extraction/compensation algorithms (2005)

50. Deng, L., Yu, D.: Use of differential cepstra as acoustic features in hidden trajectory modelling for phonetic recognition. In: Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 445–448 (2007)
51. Deng, L., Yu, D., Acero, A.: A bidirectional target filtering model of speech coarticulation: two-stage implementation for phonetic recognition. IEEE Transactions on Speech and Audio Processing **14**, 256–265 (2006)
52. Deng, L., Yu, D., Acero, A.: Structured speech modeling. IEEE Transactions on Speech and Audio Processing **14**, 1492–1504 (2006)
53. Droppo, J., Acero, A.: Noise robust speech recognition with a switching linear dynamic model. In: Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP), vol. 1, pp. I – 953–6 vol.1 (2004)
54. Fox, E., Sudderth, E., Jordan, M., Willsky, A.: Bayesian nonparametric methods for learning markov switching processes. IEEE Signal Processing Magazine **27**(6), 43 –54 (2010)
55. Frey, B., Deng, L., Acero, A., Kristjansson, T.: Algonquin: Iterating laplaces method to remove multiple types of acoustic distortion for robust speech recognition. In: Proc. European Conference on Speech Communication and Technology (EUROSPEECH) (2001)
56. Fu, Q., Zhao, Y., Juang, B.H.: Automatic speech recognition based on nonuniform error criteria. IEEE Transactions on Audio, Speech and Language Processing **20**(3), 780 –793 (2012)
57. Gales, M., Watanabe, S., Fosler-Lussier, E.: Structured discriminative models for speech recognition. IEEE Signal Processing Magazine (29), 70–81 (2012)
58. Gales, M., Young, S.: Robust continuous speech recognition using parallel model combination. IEEE Transactions on Speech and Audio Processing **4**(5), 352 –359 (1996)
59. Gao, Y., Bakis, R., Huang, J., Xiang, B.: Multistage coarticulation model combining articulatory, formant and cepstral features. In: Proc. International Conference on Spoken Language Processing (ICSLP), pp. 25–28. Beijing, China (2000)
60. Gemmeke, J., Virtanen, T., Hurmalainen, A.: Exemplar-based sparse representations for noise robust automatic speech recognition. IEEE Transactions on Audio, Speech and Language Processing **19**(7), 2067 –2080 (2011)
61. Ghahramani, Z., Hinton, G.E.: Variational learning for switching state-space models. Neural Computation **12**, 831–864 (2000)
62. Gong, Y., Illina, I., Haton, J.P.: Modeling long term variability information in mixture stochastic trajectory framework. In: In Proc. of Int. Conf on Spoken Language Processing (1996)
63. He, X., Deng, L.: DISCRIMINATIVE LEARNING FOR SPEECH RECOGNITION: Theory and Practice. Morgan and Claypool (2008)
64. He, X., Deng, L.: Speech recognition, machine translation, and speech translation —- A unified discriminative learning paradigm. IEEE Signal Processing Magazine **27**, 126–133 (2011)
65. He, X., Deng, L., Chou, W.: Discriminative learning in sequential pattern recognition — a unifying review for optimization-oriented speech recognition. IEEE Signal Processing Magazine **25**(5), 14–36 (2008)
66. Heigold, G., Ney, H., Schluter, R.: Investigations on an EM-style optimization algorithm for discriminative training of HMMs. IEEE Transactions on Audio, Speech, and Language Processing **21**(12), 2616–2626 (2013)
67. Heigold, G., Wiesler, S., Nubbaum-Thom, M., Lehnen, P., Schluter, R., Ney, H.: Discriminative HMMs. log-linear models, and CRFs: What is the difference? In: Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP) (2010)

68. Hoffman, M.D., Blei, D.M., Wang, C., Paisley, J.: Stochastic variational inference
69. Holmes, W., Russell, M.: Probabilistic-trajectory segmental HMMs. Computer Speech and Language **13**, 3–37 (1999)
70. Huang, X., Acero, A., Hon, H.W., et al.: Spoken language processing, vol. 18. Prentice Hall Englewood Cliffs (2001)
71. Huang, X., Deng, L.: An overview of modern speech recognition. In: N. Indurkhya, F.J. Damerau (eds.) Handbook of Natural Language Processing, Second Edition. CRC Press, Taylor and Francis Group, Boca Raton, FL (2010). ISBN 978-1420085921
72. Jelinek, F.: Continuous speech recognition by statistical methods. Proc. IEEE **64**(4), 532–557 (1976)
73. Juang, B.H., Hou, W., Lee, C.H.: Minimum classification error rate methods for speech recognition. IEEE Transactions on Speech and Audio Processing **5**(3), 257–265 (1997)
74. Juang, B.H., Levinson, S.E., Sondhi, M.M.: Maximum likelihood estimation for mixture multivariate stochastic observations of markov chains. IEEE International Symposium on Information Theory **32**(2), 307–309 (1986)
75. Kalinli, O., Seltzer, M., Droppo, J., Acero, A.: Noise adaptive training for robust automatic speech recognition. IEEE Transactions on Audio, Speech and Language Processing **18**(8), 1889 –1901 (2010)
76. Kalinli, O., Seltzer, M.L., Droppo, J., Acero, A.: Noise adaptive training for robust automatic speech recognition. Audio, Speech, and Language Processing, IEEE Transactions on **18**(8), 1889–1901 (2010)
77. Kello, C.T., Plaut, D.C.: A neural network model of the articulatory-acoustic forward mapping trained on recordings of articulatory parameters. Journal Acoustical Society of America **116**(4), 2354–2364 (2004)
78. King, S., J., F., K., L., E., M., K., R., M., W.: Speech production knowledge in automatic speech recognition. Journal Acoustical Society of America **121**, 723–742 (2007)
79. Kingma, D., Welling, M.: Efficient gradient-based inference through transformations between bayes nets and neural nets. In: Proc. International Conference on Machine Learning (ICML) (2014)
80. Lee, L., Attias, H., Deng, L.: Variational inference and learning for segmental switching state space models of hidden speech dynamics. In: Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP), vol. 1, pp. I–872 – I–875 (2003)
81. Lee, L.J., Fieguth, P., Deng, L.: A functional articulatory dynamic model for speech production. In: Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP), vol. 2, pp. 797–800. Salt Lake City (2001)
82. Li, J., Deng, L., Gong, Y., A.Acero: A unified framework of HMM adaptation with joint compensation of additive and convolutive distortions. Computer Speech and Language **23**, 389–405 (2009)
83. Li, J., Deng, L., Yu, D., Gong, Y., Acero, A.: High-performance hmm adaptation with joint compensation of additive and convolutive distortions via vector taylor series. In: Proc. IEEE Workshop on Automfatic Speech Recognition and Understanding (ASRU), pp. 65 –70 (2007)
84. Li, J., Deng, L., Yu, D., Gong, Y., Acero, A.: HMM adaptation using a phase-sensitive acoustic distortion model for environment-robust speech recognition. In: Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4069–4072 (2008)
85. Li, J., Deng, L., Yu, D., Gong, Y., Acero, A.: A unified framework of HMM adaptation with joint compensation of additive and convolutive distortions. Computer Speech and Language **23**(3), 389–405 (2009)

86. Liu, F.H., Stern, R.M., Huang, X., Acero, A.: Efficient cepstral normalization for robust speech recognition. In: Proc. ACL Workshop on Human Language Technologies (ACL-HLT), pp. 69–74 (1993)
87. Liu, S., Sim, K.: Temporally varying weight regression: A semi-parametric trajectory model for automatic speech recognition. IEEE Transactions on Audio, Speech and Language Processing **22**(1), 151–160 (2014)
88. Livescu, K., Fosler-Lussier, E., Metze, F.: Subword modeling for automatic speech recognition: Past, present, and emerging approaches. IEEE Signal Processing Magazine **29**(6), 44–57 (2012)
89. Ma, J., Deng, L.: A path-stack algorithm for optimizing dynamic regimes in a statistical hidden dynamic model of speech. Computer Speech and Language **14**, 101–104 (2000)
90. Ma, J., Deng, L.: Efficient decoding strategies for conversational speech recognition using a constrained nonlinear state-space model. IEEE Transactions on Audio, Speech and Language Processing **11**(6), 590–602 (2004)
91. Ma, J., Deng, L.: Target-directed mixture dynamic models for spontaneous speech recognition. IEEE Transactions on Audio and Speech Processing **12**(1), 47–58 (2004)
92. Macherey, W., Ney, H.: A comparative study on maximum entropy and discriminative training for acoustic modeling in automatic speech recognition. In: Proc. Eurospeech, pp. 493–496 (2003)
93. Mak, B., Tam, Y., Li, P.: Discriminative auditory-based features for robust speech recognition. IEEE Transactions on Speech and Audio Processing (12), 28–36 (2004)
94. Mesot, B., Barber, D.: Switching linear dynamical systems for noise robust speech recognition. IEEE Transactions on Audio, Speech and Language Processing **15**(6), 1850–1858 (2007)
95. Mnih, A., K. Gregor, .: Neural variational inference and learning in belief networks. In: Proc. International Conference on Machine Learning (ICML) (2014)
96. Moon, T.K.: The expectation-maximization algorithm. IEEE Signal Processing Magazine **13**(6), 47–60 (1996)
97. Ostendorf, M., Digalakis, V., Kimball, O.: From HMM's to segment models: A unified view of stochastic modeling for speech recognition. IEEE Transactions on Audio and Speech Processing **4**(5) (1996)
98. Ostendorf, M., Kannan, A., Kimball, O., Rohlicek, J.: Continuous word recognition based on the stochastic segment model. Proc. DARPA Workshop CSR (1992)
99. Pavlovic, V., Frey, B., Huang, T.: Variational learning in mixed-state dynamic graphical models. In: UAI, pp. 522–530. Stockholm (1999)
100. Picone, J., Pike, S., Regan, R., Kamm, T., bridle, J., Deng, L., Ma, Z., Richards, H., Schuster, M.: Initial evaluation of hidden dynamic models on conversational speech. In: Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP) (1999)
101. Povey, D., Kanevsky, D., Kingsbury, B., Ramabhadran, B., Saon, G., Visweswariah, K.: Boosted MMI for model and feature-space discriminative training. In: Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4057–4060 (2008)
102. Povey, D., Kingsbury, B., Mangu, L., Saon, G., Soltau, H., Zweig, G.: fMPE: Discriminatively trained features for speech recognition. In: Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP), vol. 1, pp. 961–964 (2005)

103. Povey, D., Woodland, P.C.: Minimum phone error and I-smoothing for improved discriminative training. In: Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP), vol. 1, pp. I–105 (2002)
104. Povey, D., Woodland, P.C.: Minimum phone error and i-smoothing for improved discriminative training. In: Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 105–108 (2002)
105. Rabiner, L.: A tutorial on hidden markov models and selected applications in speech recognition. Proceedings of the IEEE **77**(2), 257–286 (1989)
106. Rabiner, L., Juang, B.H.: An introduction to hidden markov models. IEEE ASSP Magazine **3**(1), 4–16 (1986)
107. Rabiner, L., Juang, B.H.: Fundamentals of Speech Recognition. Prentice-Hall, Upper Saddle River, NJ. (1993)
108. Rosti, A., Gales, M.: Rao-blackwellised gibbs sampling for switching linear dynamical systems. In: Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP), vol. 1, pp. I – 809–12 vol.1 (2004)
109. Russell, M., Jackson, P.: A multiple-level linear/linear segmental HMM with a formant-based intermediate layer. Computer Speech and Language **19**, 205–225 (2005)
110. Sakoe, H., Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. In: Readings in Speech Recognition, pp. 159–165. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1990)
111. Schlueter, R., Macherey, W., Mueller, B., Ney, H.: Comparison of discriminative training criteria and optimization methods for speech recognition. Speech Communication **31**, 287–310 (2001)
112. Seide, F., Li, G., Yu, D.: Conversational speech transcription using context-dependent deep neural networks. In: Proc. Annual Conference of International Speech Communication Association (INTERSPEECH), pp. 437–440 (2011)
113. Sun, J., Deng, L.: An overlapping-feature based phonological model incorporating linguistic constraints: Applications to speech recognition. Journal Acoustical Society of America **111**, 1086–1101 (2002)
114. Suzuki, J., Fujino, A., Isozaki, H.: Semi-supervised structured output learning based on a hybrid generative and discriminative approach. In: Proc. EMNLP-CoNLL (2007)
115. Wang, Y., Gales, M.J.: Speaker and noise factorization for robust speech recognition. IEEE Transactions on Audio, Speech and Language Processing **20**(7), 2149–2158 (2012)
116. Woodland, P.C., Povey, D.: Large scale discriminative training of hidden markov models for speech recognition. Computer Speech and Language (2002)
117. Wright, S., Kanevsky, D., Deng, L., He, X., Heigold, G., Li, H.: Optimization algorithms and applications for speech and language processing. IEEE Transactions on Audio, Speech, and Language Processing **21**(11), 2231–2243 (2013)
118. Xing, E., Jordan, M., Russell, S.: A generalized mean field algorithm for variational inference in exponential families. In: Proc. Uncertainty in Artificial Intelligence (2003)
119. Yu, D., Deng, L.: Speaker-adaptive learning of resonance targets in a hidden trajectory model of speech coarticulation. Computer Speech and Language **27**, 72–87 (2007)
120. Yu, D., Deng, L., Acero, A.: A lattice search technique for a long-contextual-span hidden trajectory model of speech. Speech Communication **48**, 1214–1226 (2006)
121. Yu, D., Deng, L., Dahl, G.: Roles of pre-training and fine-tuning in context-dependent DBN-HMMs for real-world speech recognition. In: NIPS Workshop on Deep Learning and Unsupervised Feature Learning (2010)

122. Yu, D., Deng, L., Gong, Y., Acero, A.: A novel framework and training algorithm for variable-parameter hidden markov models. IEEE Transactions on Audio, Speech and Language Processing **17**(7), 1348–1360 (2009)
123. Yu, D., Deng, L., He, X., Acero, A.: Use of incrementally regulated discriminative margins in MCE training for speech recognition. In: Proc. Annual Conference of International Speech Communication Association (INTERSPEECH) (2006)
124. Yu, D., Deng, L., He, X., Acero, A.: Use of incrementally regulated discriminative margins in mce training for speech recognition. In: Proc. International Conference on Spoken Language Processing (ICSLP), pp. 2418–2421 (2006)
125. Yu, D., Deng, L., He, X., Acero, A.: Large-margin minimum classification error training: A theoretical risk minimization perspective. Computer Speech and Language **22**, 415–429 (2008)
126. Zen, H., Tokuda, K., Kitamura, T.: An introduction of trajectory model into HMM-based speech synthesis. In: in Proc. of ISCA SSW5, pp. 191–196 (2004)
127. Zhang, B., Matsoukas, S., Schwartz, R.: Discriminatively trained region dependent feature transforms for speech recognition. In: Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP), vol. 1, pp. I–I (2006)
128. Zhang, L., Renals, S.: Acoustic-articulatory modelling with the trajectory HMM. IEEE Signal Processing Letters **15**, 245–248 (2008)
129. Zhang, S., Gales, M.: Structured SVMs for automatic speech recognition. IEEE Transactions on Audio, Speech and Language Processing **21**(3), 544 –555 (2013)
130. Zhou, J.L., Seide, F., Deng, L.: Coarticulation modeling by embedding a target-directed hidden trajectory model into HMM — model and training. In: Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP), vol. 1, pp. 744–747. Hongkong (2003)