Computer Speech and Language (1993) 2, 101–119

Hidden Markov models with templates as non-stationary states: an application to speech recognition

Oded Ghitza and M. Mohan Sondhi

AT&T Bell Laboratories, Acoustics Research Department, Murray Hill, New Jersey 07974, U.S.A.

Abstract

In most implementations of hidden Markov models (HMMs) a state is assumed to be a stationary random sequence of observation vectors whose mean and covariance are estimated. Successive observations in a state are assumed to be independent and identically distributed. These assumptions are reasonable when each state represents a short segment of the speech signal. When states represent longer portions of the signal (e.g. phonemes, diphones, etc.) both assumptions are inaccurate. Recently, some attempts have been made to incorporate correlations between successive observations in a state. But to our knowledge, non-stationarity has not been dealt with. We propose an alternative representation in which a state of an HMM is defined as a template, i.e. a "typical" sequence of observations. The template for a state is derived from an ensemble of segments corresponding to that state. In our present implementation, the observations are 11th-order cepstrum vectors plus energy, states represent diphones and ensembles of the diphones are obtained from a hand-labeled speaker-dependent database of 2000 sentences spoken fluently. The probability of a test sequence being generated in a given state is obtained by time-warping the test utterance to the template, and assuming the differences between the corresponding observations to have a joint distribution. Tests on 50 sentences (outside the training set) indicate a correct recognition rate for phonemes of about 70%.

1. Introduction

Consider a Markov chain with N states, $Q \equiv [q_1, q_2, \ldots, q_N]$, and associated transition probability matrix $A \equiv [a_{ij}, 1 \leqslant i,j \leqslant N]$. If S_k denotes the state of the Markov chain at time instant k, then by definition $a_{ij} = \text{prob}(S_{k+1} = q_j | S_k = q_i)$. A hidden Markov model (HMM) based on this Markov chain generates a random sequence of observation vectors $\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_k, \ldots$, whose statistical properties change as the state of the underlying Markov chain changes. In order to clarify our notation, we show an illustrative example of observations generated by an HMM, in Fig. 1. The index of S and $\mathbf{o}$ is the running time index. The index of q indicates the particular choice from the set Q. Thus, the Markov chain stayed in the state q_6 for the first nine time instants. During this

0885–2308/93/020101 + 19 $08.00/0

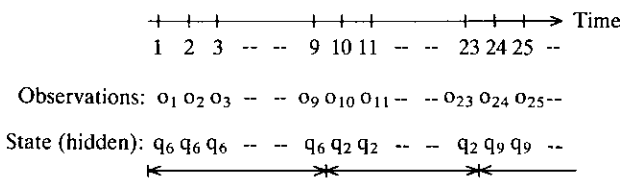

Figure 1. An illustration of observations generated by a hidden Markov model.

time interval, the sequence of observations $\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_9$ was generated with a probability distribution appropriate to the state q_6, etc.

In almost all applications of HMMs to speech recognition, the probability distribution of the observation $\mathbf{o}_k$, generated at time instant k, is assumed to depend only on the state $S_k \in Q$, in which it is generated. Hence, the observations generated in any given state are independent and identically distributed (i.i.d.). Thus, if the **sequence** of observations $\mathbf{O} \equiv [\mathbf{o}_t, \mathbf{o}_{t+1}, \ldots, \mathbf{o}_{t+T-1}]$ is generated in some state q (i.e. if $S_t = S_{t+1} = \ldots = S_{t+T-1} = q$), then the assumption is that the probability of that sequence has the form:

$$P(\mathbf{O}) = \prod_{k=t}^{t+T-1} p(\mathbf{o}_k | q). \tag{1}$$

The state-dependent probability distribution $p(\mathbf{o}|q)$ can take a variety of forms. If the observations are p-dimensional vectors of continuously distributed components, the distribution is usually assumed to be a p-dimensional Gaussian distribution (or a mixture of such distributions).

Some more general models have been considered in the literature (although not as widely used). Thus in Bahl, Jelinek and Mercer (1983), $\mathbf{o}_k$ is assumed to depend on S_k as well as on the previous state, S_{k-1}. In Wellekens (1987), $\mathbf{o}_k$ is allowed to depend on S_k, S_{k-1} and $\mathbf{o}_{k-1}$, i.e. on the previous observation as well.

Even with these generalizations, a sequence of observations generated in a given state is a segment of a stationary time-discrete random process. In certain situations (e.g. for a state representing the middle portion of a steady vowel), this assumption of stationarity is reasonable. If, however, the state is to represent a plosive, or a long segment of speech (longer than 30 or 40 msec, say) the assumption is clearly invalid. To the best of our knowledge, no one has considered HMMs in which the states are non-stationary, i.e. in which the probability of an observation sequence depends **explicitly** on the time index, k. It is this extension that is the subject of the present paper.

Our motivation for studying such a model comes from the application of HMMs to speech recognition in terms of sub-word units. Such HMMs are of interest in large-vocabulary recognition, as well as in other applications where a decoding in terms of sub-word units is desirable. Specifically, consider the HMM "phonetic decoder" presented by Levinson (1986, 1987), in which each state represents a (variable-duration) phone. With this choice of sub-word units, the model has about 50 states, each specified by a probability density for the duration, and a probability density for the observations. Successive observations in a state are assumed, as above, to be i.i.d. Let us consider the problem faced by this model in representing the spoken sentence "I have a feeling" whose spectrogram is shown in Fig. 2. Shown below the spectrogram is an approximate

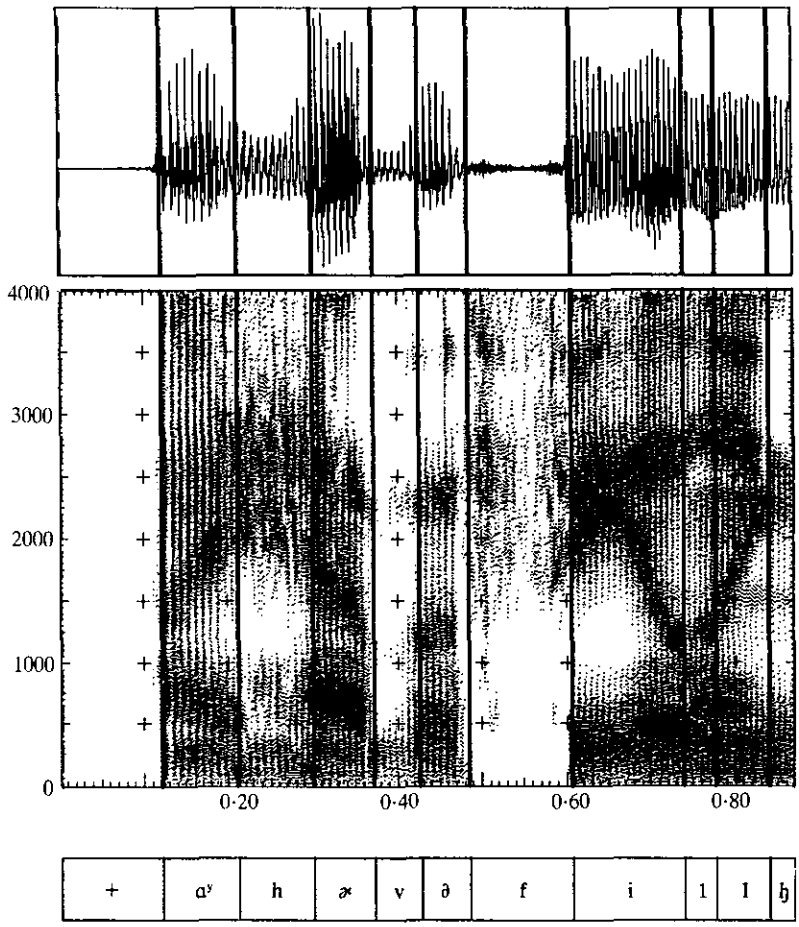

Figure 2. A spectrogram of the sentence "I have a feeling". Notice, for example, the non-stationarity of the vowel (i). (See Table I for the definition of the phonetic symbols.)

phonetic transcription. It is clear that if the phone [i], say, is represented by a state in the HMM, that state must be non-stationary. (In fluent speech such non-stationary states are the rule, and "steady" states the rare exception.) To represent such a state by time-averaged statistical properties is a gross approximation. Another unsatisfactory feature is that because of the i.i.d. assumption, the probability assigned to a set of observations is independent of the order in which the observations occur. Thus, for instance, reversing the direction of the formant transitions leaves the probability unchanged.

The way this non-stationarity has been dealt with in the past is by representing the transient state as a concatenation of two or more sub-states. Thus, the non-stationary state is approximated by a sequence of piecewise stationary states. In principle, any transient state can be approximated this way by a sufficiently fine subdivision. However, such subdivision cannot model the statistical dependence. If a long sequence of dependent observations is supposed to be generated by a chain of sub-states, then the dependence extends over all those sub-states.

We propose an alternative point of view in which the entire sub-word unit is regarded as a single **non-stationary** state. Our representation combines features of dynamic time warping (DTW) and HMM. We define a state by a fixed length template, i.e. a "typical" sequence of observations. The template for each state is derived from a labeled database, much like a word template in a DTW word recognizer. To compute the probability of generating a test sequence in a given state the test sequence is first time-warped to the template. This warped sequence is assumed to be drawn from a distribution whose mean is the template. The vector differences, ε_k, between the aligned observations are assumed to have a joint Gaussian distribution. This representation can handle non-stationarity as well as statistical dependence of observation sequences generated in the state. For the present, we have assumed the deviations ε_k to be uncorrelated, so that the covariance matrix is block diagonal. Note, however, that the template specifies a non-stationary mean; also we allow the covariance matrix to be time-varying. A complete description of the way in which we estimate the statistical properties for each state is given in the next section.

A moment's reflection shows that a diphone is the smallest sub-word unit for which such an HMM with non-stationary states makes sense. This is because of co-articulation. The spectral trajectory of, say, the vowel [ɑ] is quite different in the CV syllable /bo/ from that in the syllable /go/, as shown in Fig. 3. A similar observation can be made with respect to the effect of right context. Clearly, a model for the phoneme [ɑ] derived from occurrences of [ɑ] in all contexts would be noisy due to co-articulation. Even if several models were estimated, depending on the left (right) context, the right (left) half of the model would still be noisy. From the very outset, therefore, we consider states to represent diphones. By defining a diphone from the midpoint of the first phoneme to the

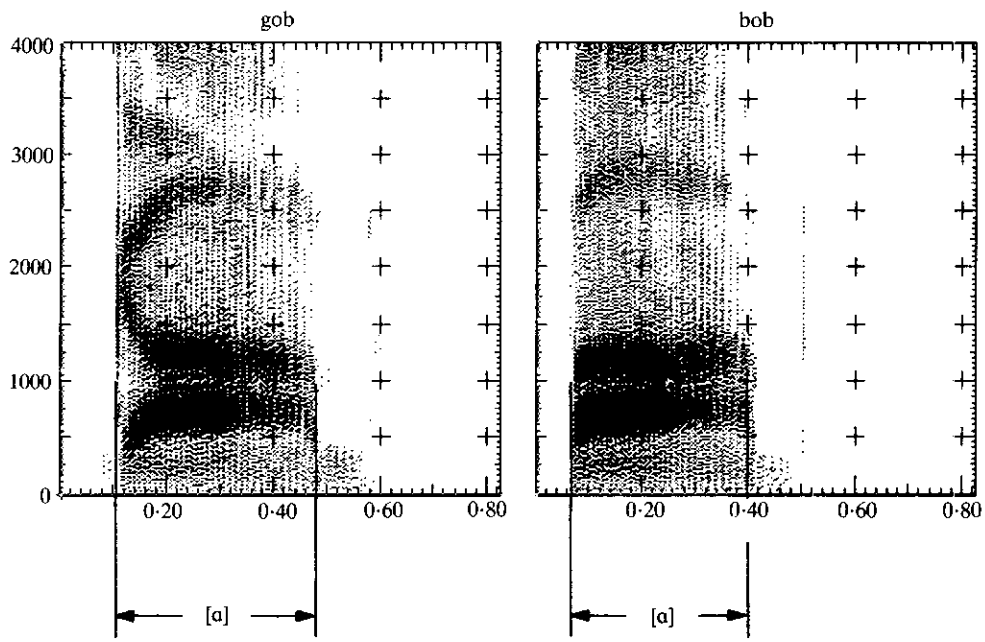

Figure 3. Example of variation of the vowel [ɑ] due to co-articulation.

midpoint of the second, we minimize the effect of co-articulation due to both the right and the left context. (It is, of course, possible to consider even more complicated sub-word units. However, we have not done that.)

To the best of our knowledge, HMMs with non-stationary states have not been proposed in the literature so far.[1] However, representation of phones as sequences of observations, or stochastic segments, has been proposed recently by Roucos and Dunham (1987) and Ostendorf and Roukos (1989). There are several crucial differences between their approach and ours. The first major difference is, of course, that unlike our implementation, they do not consider an HMM framework for the segments. Thus, the probability of a segment is independent of the preceding segment. A second important difference is in the choice of sub-word units. Instead of diphones their implementation uses phones. As mentioned in the preceding paragraph, the diphone is the smallest unit for which we expect to see significant advantage of this approach. Also, the choice of diphone as the unit imposes a structure on the transition probability matrix, which can be used to advantage (see Section 2.4). The third difference is in the method of alignment of test segments to templates. We use dynamic time warping, while they use trace segmentation or linear time warping, to compensate for variability in speaking rate. As is well known, variations in speaking rate have a much larger effect on the durations of fricatives and steady portions of vowels than on the durations of transient portions and plosives. In general, therefore, linear warping is inferior to DTW, because it scales these variations uniformly.

The rest of the paper is organized as follows: in the next section we give a description of our proposed HMM with non-stationary states; in Section 3, we present the results of a preliminary recognition experiment using such a model; in Section 4, we discuss various ways in which the current model should be modified and extended.

2. Description of the HMM

The structure of our HMM is similar to that of the variable duration HMM described by Levinson (1986). The main difference is, of course, in the definition of a state, and in the manner in which a probability is assigned to a sequence generated in a given state.

As mentioned in the Introduction, we have chosen the states to be diphones. Assuming there are about 50 phonemes in English, the upper bound on the number of states, N, is about 2500. In practice, $N = 1000$ should suffice (e.g. Lee, *et al.*, 1990).

The dwell time in a state of a conventional HMM is exponentially distributed. As this is not, in general, a good approximation to the duration distribution, we replace the underlying Markov chain by a semi-Markov chain, as in Levinson (1986). Thus, the $N \times N$ state transition matrix A is constrained to have its diagonal elements $a_{ii} = 0$, for all i, and the dwell time in a state is governed by a state-dependent probability distribution of durations.

The definition of a state is in terms of a template (or typical sequence of observations) and a probability distribution of the deviations from the template. We turn now to the procedures by which the templates, the probability distributions and the state transition probabilities are determined. We begin with a description of the database used for training.

[1] A reviewer points out that a paper in press (Li Deng) also deals with non-stationarity of states in an HMM.

2.1. Database and analysis conditions

To train the HMM outlined in the previous section, we used a database supplied to us by J. P. Olive. This database, which will be described in detail in Section 3, consists of speech sentences spoken fluently by one male speaker. A large part of this database has been meticulously hand labeled by Olive, so as to indicate, on the speech waveform, the beginning, middle and end points of phonemes. The set of phonemes, and the symbols used to represent them are shown in Table I. From the labeled phonemes it was a simple matter to collect an ensemble of all occurrences of any selected diphone, where we define a diphone $p_1\_p_2$ to be the waveform from the middle of the first phoneme p_1 to the middle of the second phoneme p_2.

The speech was recorded in a studio, and digitized to 16-bit samples at a sampling rate of 8000 samples/s. An LPC-based cepstrum analysis was performed on each sentence in the database. The LPC order was 12; the order of the cepstrum vectors obtained from these LPC vectors was 11. The window length was 30 ms, and the overlap between successive windows was 10 ms. The observation vectors (corresponding to the $\mathbf{o}_k$ of Fig. 1) in our study are 12-dimensional vectors comprised of the 11-dimensional cepstrum vectors, plus one component representing short-term energy, computed over the 30 ms windows.[2] A token for the diphone $p_1\_p_2$ is the sequence of observation vectors located between the midpoints of the phonemes p_1 and p_2. (For speaker-independent recognition the cepstrum vector is often liftered. This reduces the variability due to the tilt in the spectrum of the glottal excitation or of the transmission medium. However, for the speaker-dependent case under study here, we found in a preliminary study that liftering does not help. So we have not used a lifter in this study.)

TABLE I. Set of phonemes: symbols and examples

Symbol	Example	Symbol	Example	Symbol	Example
ɛ	bet	l	led	m	mom
ɔ	bought	r	red	n	nun
ɑ	cot	w	wet	ŋ	sing
u	boot	y	yet	p	pop
ɚ	bird	h	hay	t	tot
ɑʸ	bite	s	sister	k	kick
eʸ	bait	ʃ	shoe	p⁻	p closure
aʷ	now	z	zoo	t⁻	t closure
ə	schwa	ʒ	measure	k⁻	k closure
ɪ	bit	č	church	b	bob
æ	bat	ǰ	judge	d	dad
ʌ	butt	ɵ	thief	g	gag
ʊ	book	ð	they	b⁻	b closure
ɔʸ	boy	f	fief	d⁻	d closure
i	beat	v	verve	g⁻	g closure
oʷ	boat				

[2] In one of the experiments (see Section 3.2) the energy was averaged over 10 ms windows.

2.2. Derivation of the templates

In order not to clutter the notation, we will describe the procedure for some selected state q. The same procedure is followed for every state. Let $\mathbf{O}^i$, $1 \leqslant i \leqslant I$, be the I observation sequences comprising the ensemble of tokens found in the database, for the selected state, q. Let T_i be the length of $\mathbf{O}^i$, i.e. the number of observation vectors in $\mathbf{O}^i$. Let $D(\mathbf{O}^i,\mathbf{O}^j)$ be the distance (to be defined shortly) between $\mathbf{O}^i$ and $\mathbf{O}^j$. Then, following the modified k-means method (Wilpon & Rabiner, 1985), we define the template for the state q as the observation sequence whose cumulative distance from all other sequences in the ensemble is a minimum, i.e. the sequence $\bar{\mathbf{O}}$ such that:

$$\sum_{i=1}^{I} D(\bar{\mathbf{O}},\mathbf{O}^i) \leqslant \sum_{i=1}^{I} D(\mathbf{O}^j,\mathbf{O}^i), \quad \text{for all } j. \tag{2}$$

In the unlikely event that several sequences have this property, any of them may be arbitrarily selected to be the template.

We have used the term "distance" for the function D in Equation (2) in the loose sense in which it is used in the speech recognition literature. Indeed, for the problem at hand, no useful function that is a true metric has ever been proposed. The reason is that although all the observation vectors have the same dimension, variable-length sequences of these vectors have variable dimension. Distance metrics may be defined for such sets, but they tend to be artificial and not very useful for speech signals. Fortunately, it is not necessary for D to be a true metric. It need only be some reasonable measure of dissimilarity of the two sequences of vectors.

We define the distance D between sequences as the one used in the DTW method of speech recognition. As a first step towards defining D, we need to define the distance $d(\mathbf{a},\mathbf{b})$, between any two vectors, $\mathbf{a}$ and $\mathbf{b}$. When the observations are cepstrum vectors, a suitable definition for d is:

$$d(\mathbf{a},\mathbf{b}) = (\mathbf{a}-\mathbf{b})'\Sigma^{-1}(\mathbf{a},\mathbf{b}). \tag{3}$$

Here $'$ denotes vector transpose, and Σ is a diagonal matrix whose ith diagonal entry is the variance of the ith cepstral coefficient.

In terms of the distance d, we can define the distance between $\mathbf{O}^i$ and $\mathbf{O}^j$ by the usual DTW procedure. Let $\mathbf{o}^i_m$, $m = 1, 2, \ldots, T_i$ be the vectors in sequence $\mathbf{O}^i$, and $\mathbf{o}^j_n$, $n = 1, 2, \ldots, T_j$ the vectors in sequence $\mathbf{O}^j$. Then we define $D(\mathbf{O}^i,\mathbf{O}^j)$ as:

$$D(\mathbf{O}^i,\mathbf{O}^j) = \frac{1}{T_i} \min_{n(m)} \sum_{m=1}^{T_i} d(\mathbf{o}^i_m,\mathbf{o}^j_{n(m)}). \tag{4}$$

The mapping $n(m)$ is constrained such that $n(1) = 1$ and $n(T_i) = T_j$. Thus, $D(\mathbf{O}^i,\mathbf{O}^j)$ is the average distance between corresponding observation vectors in the two sequences, after the sequence $\mathbf{O}^j$ has been optimally warped on to the sequence $\mathbf{O}^i$. The search for the optimal map is done, as usual, by a dynamic programming algorithm. In order to minimize the effects of time quantization, both sequences are upsampled (by linear interpolation), and they are both linearly warped to the same length before the DTW. Also, as is usually done to avoid pathological mappings, the local slope of the map is restricted to be between 0·5 and 2·0.

Note that the definition of D in Equation (4) is not symmetric in its arguments. However, in the present work, we always need the distances of various sequences from a distinguished sequence. Therefore, we do not symmetrize the distance.

The minimization required to find the template according to Equation (2) is done by exhaustive search. Thus, a particular sequence is selected, and the sum of its distances to all other members of the ensemble is computed by using Equation (4). The process is repeated for every member of the ensemble, and the one for which the sum is a minimum, is chosen to be the template for the state q. As a mnemonic, let us denote the length of $\bar{\mathbf{O}}$ by $\bar{T}$.

Once the template $\bar{\mathbf{O}}$ of length $\bar{T}$ has been derived, we can derive a covariance matrix $\mathbf{\Phi}$ for the state. To do this, let us warp each token $\mathbf{O}^i$ in the training data for the state, to the template $\bar{\mathbf{O}}$, using the distance function defined in Equation (4). Let $\hat{\mathbf{O}}^i \equiv [\mathbf{o}^i_{t_1}, \mathbf{o}^i_{t_2}, \ldots, \mathbf{o}^i_{t_{\bar{T}}}]$ be the time-warped sequence. Then we define $\mathbf{\Phi}$ to be a $\bar{T} \times \bar{T}$ block matrix, in which the mnth block is a $p \times p$ matrix:

$$\mathbf{\Phi}_{mn} = \frac{1}{I} \sum_{i=1}^{I} (\mathbf{o}^i_{t_m} - \bar{\mathbf{o}}_m)(\mathbf{o}^i_{t_n} - \bar{\mathbf{o}}_n)', \tag{5}$$

where I is the number of tokens for the state available in the database.

In the next section it will be convenient to view $\mathbf{\Phi}$ in a different way. Suppose we define a long vector $\bar{V}$ obtained by stacking (concatenating) the components of the observation vectors of $\bar{\mathbf{O}}$. Thus, $\bar{V}$ is of dimension $p\bar{T}$. Similarly, let us define $\hat{V}^i$ by stacking the observations of $\hat{\mathbf{O}}^i$. Then, clearly, an alternative representation for $\mathbf{\Phi}$ is:

$$\mathbf{\Phi} = \frac{1}{I} \sum_{i=1}^{I} (\hat{V}^i - \bar{V})(\hat{V}^i - \bar{V})'. \tag{5a}$$

2.3. Probability density of a sequence conditioned on state

Let $\mathbf{O} \equiv [\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_T]$ be a given sequence of observations. Given that $\mathbf{O}$ was generated in the state q, what is its conditional probability density $p(\mathbf{O}, T | \text{state} = q)$?

We will assign a meaning to this probability density only for the case where the sequence is the entire sequence generated in the state q, i.e. assuming the source enters the state q with the observation $\mathbf{o}_1$ and exits to some other state after generating the sequence $\mathbf{O}$. With this proviso, the probability density is a product of two factors, the probability that the sequence has duration T, and the probability that the observation sequence is the one specified. Thus:

$$p(\mathbf{O}, T | q) = p(\mathbf{O} | T, q) p(\text{dur} = T | q). \tag{6}$$

In order to define the first factor on the right-hand side of Equation (6), we again use dynamic time warping. Let $\bar{\mathbf{O}} \equiv [\bar{\mathbf{o}}_1, \bar{\mathbf{o}}_2, \ldots, \bar{\mathbf{o}}_{\bar{T}}]$ be the template for the state q. Let $\hat{\mathbf{O}}$ be the sequence of $\bar{T}$ vectors obtained by warping $\mathbf{O}$ to the template $\bar{\mathbf{O}}$. Then we postulate that the components of $\hat{V}$ are jointly Gaussian. (An extension to Gaussian mixtures is, of course, possible.) The mean, $\boldsymbol{\mu}$, of the Gaussian distribution is defined to be the vector $\bar{V}$ obtained by similarly stacking the $\bar{T}$ observation vectors of the template, $\bar{\mathbf{O}}$. The covariance matrix, $\mathbf{\Phi}$, is the matrix $\mathbf{\Phi}$ defined in the previous section. Note that we view $\mathbf{\Phi}$ here as a $p\bar{T} \times p\bar{T}$ matrix, rather than as a block matrix.

Although $\mathbf{\Phi}$ can be a full covariance matrix, we have so far restricted it to be block diagonal, i.e. $\Phi_{ij}=0$, $i\neq j$. This means that we assume the deviations of the observation vectors from their means at any pair of time instants to be statistically independent. However, we have experimented with various constrained forms for the diagonal blocks. We have considered the diagonal blocks to be full covariance matrices, or diagonal matrices. We have considered the blocks to be identical, or to have two possible values—one for the first phoneme, p_1, and a different one for the second phoneme, p_2, of the diphone $p_1\_p_2$ corresponding to the state, etc. We will discuss the effects of such choices in Section 3.

As for the duration probability density, we have for the present replaced it by a penalty function which has the value 1 over the allowed range of durations, and 0 outside. The allowable range is taken to be $\gamma_1\bar{T}\leqslant T\leqslant\gamma_2\bar{T}$. Typically, we choose $\gamma_1=0{\cdot}5$ and $\gamma_2=2{\cdot}0$.

In summary, if $g(.,\boldsymbol{\mu},\mathbf{\Phi})$ is a $p\bar{T}$-dimensional Gaussian density, with mean vector $\boldsymbol{\mu}$ and covariance matrix $\mathbf{\Phi}$, then we define:

$$p(\mathbf{O},T|q)=g(\hat{V},\boldsymbol{\mu},\mathbf{\Phi}),\quad \gamma_1\bar{T}\leqslant T\leqslant\gamma_2\bar{T}$$

$$=0,\quad \text{otherwise,} \tag{7}$$

where $\hat{V}$ is the stacked vector defined above.

2.4. The state transition probabilities

In the rest of the paper we will introduce a slight change in the meaning of the notation S_k, to reflect the difference between the traditional definition of a state and our state. S_k will still denote some state drawn from the set Q. However, instead of denoting the state of the Markov chain at time instant k, it will denote the kth state in some sequence of non-stationary states. While in this state, the source might generate τ_k observations, in which case the time index advances by τ_k time steps.

The fact that we have chosen diphones as our states, implies a certain inherent syntactic constraint on possible state sequences, quite apart from any additional grammatical constraints that might be imposed. Thus, if the diphone represented by state q_i is $p_1\_p_2$ and the diphone represented by state q_j is $p_3\_p_4$, then:

$$a_{ij}=\text{prob}(S_{k+1}=q_j|S_k=q_i)$$

$$=0,\ \text{unless } p_2\equiv p_3. \tag{8}$$

For the state transitions which satisfy the constraint $p_2\equiv p_3$, it is possible to estimate the transition probabilities from the database. However, once the probabilities of the forbidden transitions has been set to 0, we conjecture that it is not important to estimate the exact probabilities of the remaining transitions. For the present we therefore assume that all "legal" transitions from a state are equally probable.

2.5. Decoding a test utterance

In this section we describe the procedure used to decode an unknown test utterance $\mathbf{O}$,

assuming that all the templates and covariance matrices for the states have already been derived. By decoding we mean inferring the sequence of states of the underlying Markov chain, which generated the given sequence **O**. We do this by finding the maximum likelihood segmentation of the sequence.

To explain our procedure, suppose we postulate a segmentation of **O** into M contiguous segments $\mathbf{O}_1, \mathbf{O}_2, \ldots, \mathbf{O}_M$. Let $\tau = (\tau_1, \tau_2, \ldots, \tau_M)$ be the vector of durations of these segments, and let $\mathbf{S} = S_1, S_2, \ldots, S_M$ be the corresponding sequence of states in which the segments were generated. Also, let $S_k = q_{i_k} \in Q$. Then, for this hypothesized segmentation, the log likelihood L may be written as:

$$L(\mathbf{S},\tau) = \sum_{k=1}^{M} [\log a_{i_{k-1}i_k} + (\tau_k / \bar{T}_{i_k}) \log p(\mathbf{O}_k, \tau_k | q_{i_k})], \tag{9}$$

where the probability $p(.)$ is given in Equation (7).[3] This is the usual definition of log likelihood, except for the normalization factor $(\tau_k / \bar{T}_{i_k})$. Our rationale for using this factor is as follows: the quantity $p(.)$ is a joint Gaussian distribution of $\bar{T}_k$ vectors. Thus, division by $\bar{T}_k$ gives the log likelihood per observation vector. And multiplication by τ_k gives the total log likelihood for the τ_κ observations in the segment. We tried several modifications of this normalization factor, but this one gave the best performance.

The decoding is obtained by finding $(\mathbf{S},\tau)$ which maximize L. (For speech recognition only the state sequence is of interest, although, of course, the maximization must be over both **S** and τ. For applications to segmentation, the durations too are of interest.)

The maximization can be performed by a dynamic program which is quite analogous to that given in Levinson (1986), so we will not detail it here. Note, that in view of Equations (7) and (8), only the allowed durations and transitions need be searched.

3. Experimental results

In this section we will describe the results of some recognition experiments which we have conducted with recognizers of the type described in the previous sections. We conducted two sets of experiments using two different databases. In the first set of experiments, described in Section 3.1, we compared the performance of the proposed HMM to that of a traditional HMM using the same database (termed KBDY) and analysis conditions. In Section 3.2 we describe a second set of experiments, aimed at studying the effects of various assumptions of our model. These experiments were conducted on a different database (termed KBB).

Each of the databases consists of some 2000 sentences, spoken fluently by the same male speaker (K. B. Bauer). We were forced to use two different databases for a practical reason: in the initial stage of our study, only database KBDY was available. Database KBB, which has a much larger number of labeled segments, was generated only at a later stage. Unfortunately, the recording conditions for the two databases, though similar, were not identical. This prevented us from merging the two when database KBB became available.

The 2000 sentences of the KBB database include the 450 TIMIT sentences, about 470 sentences composed to cover all dyads, and about 80 long sentences incorporating syntactic and prosodic variations.

[3] For $k = 1$, the term $a_{i_0 i_1}$ is to be interpreted as the probability of initially choosing the state q_{i_1}.

Both databases were processed similarly, using the analysis conditions described in Section 2.1. Templates, for both databases, were derived in the manner described in Section 2.2.

3.1. Preliminary experiment

In this experiment we compared the performance of our recognizer with the performance of a base-line recognizer, using database KBDY. The method we used to measure the performance of our recognizer is discussed first, followed by the results for the base-line system.

We decided to consider only those diphones for which we found at least three tokens in the database. For the KBDY database, we have obtained such ensembles for 976 different diphones. The number of tokens in any such ensemble is between 3 and 47, with an average of 16. For each of these ensembles we derived a template in the manner described in Section 2.2. The observation vectors, which were derived in the manner described in Section 2.1, included energy which was averaged over a 30 ms window.

For the probability distribution, as mentioned in Section 2.3, we assume the covariance matrix $\mathbf{\Phi}$ to be block diagonal. However, we estimated full covariance matrices for the diagonal blocks. For 274 of the 976 states, we estimated two state-dependent matrices, one for the first phoneme and one for the second phoneme of the diphone. For the remaining states we did not have an adequate number of tokens to derive state-dependent covariance matrices. For these states we used a "prototype" matrix for each phoneme in the diphone. The prototypes were computed by pooling tokens of structurally similar diphones. Thus, two matrices were computed from the pooled data for all vowel–vowel diphones, two from the pooled data for all vowel–consonant diphones, etc.

Once all the states had been derived, the model was tested on a set of 30 sentences which were not part of the training data. These sentences were also hand-labeled by J. P. Olive. To test the accuracy of the transcriptions, the recognized sequence of diphones was first (trivially) converted to a string of phonemes by applying the syntactic constraint mentioned in Section 2.4. Each recognized phoneme string was aligned to the corresponding hand-labeled string for minimum Levenshtein distance. (That is, the alignment that minimized the total number of deletions + substitutions + insertions.) From the aligned sequences we determined the number of correctly recognized phonemes, C, the number of insertions, I, and the number of deletions, D. The recognizer may be scored as follows:

$$\% \text{ correct} = 100 \times C/R \tag{10a}$$

$$\% \text{ insert} = 100 \times I/R \tag{10b}$$

$$\% \text{ delet} = 100 \times D/R, \tag{10c}$$

where R is the total number of phonemes in the test sentences. For the 30 test sentences, $R = 518$.

Before presenting the scores obtained, we must address one complication that arises due to diphones that appear in the test data, but for which we have not yet derived states. In the test sentences, "schwa" and "R" appear a total of 67 times; however, these

phonemes were not labeled in the database. So we do not have states corresponding to the diphones involving these phonemes. We can do one of three things: (i) assume these missing diphones would have been all incorrectly recognized; (ii) ignore the missing diphones altogether (i.e. assume they were not present in the input string); or (iii) assume they would all have been correctly recognized. The top rows on the left-hand side of Tables II, III and IV, respectively, show the scores obtained under each of these assumptions.

It would, of course, be preferable to derive a state for "schwa" and show a single result instead of three. There is, however, a genuine difficulty. The reason why "schwa" was not labeled in the training data is that it *cannot* be labeled reliably in fluent speech. The same is true of several other ambiguities in fluent speech which we will discuss later, in Section 3.3. There we will argue that the segmentation should be in terms of "equivalence classes", and that the ambiguities within the classes should be resolved on the basis of context, grammatical constraints, etc. As will be seen in Section 3.3, with that modification, we no longer need special consideration for unlabeled states.

TABLE II. New HMM vs. base-line HMM, "R" and "schwa" assumed wrong, database KBDY

	Equation (10)			Equation (11)		
	Per cent correct	Per cent insert	Per cent delet	Per cent correct	Per cent insert	Per cent delet
New HMM	61·0	20·5	3·7	50·6	17·0	3·0
Base-line HMM	55·4	48·8	3·1	37·2	32·8	2·1

TABLE III. New HMM vs. base-line HMM, "R" and "schwa" ignored, database KBDY

	Equation (10)			Equation (11)		
	Per cent correct	Per cent insert	Per cent delet	Per cent correct	Per cent insert	Per cent delet
New HMM	70·1	23·5	4·2	56·7	19·0	3·4
Base-line HMM	63·6	56·1	3·5	40·8	35·9	2·3

TABLE IV. New HMM vs. base-line HMM, "R" and "schwa" assumed correct, database KBDY

	Equation (10)			Equation (11)		
	Per cent correct	Per cent insert	Per cent delet	Per cent correct	Per cent insert	Per cent delet
New HMM	73·9	20·5	3·7	61·4	17·0	3·0
Base-line HMM	68·3	48·8	3·1	45·9	32·8	2·1

The percentage of correctly recognized phonemes given by Equation (10a) is a reasonable measure of performance if one assumes that insertions can be eliminated by the lexical access stage of the recognizer. However, it is clearly not an accurate reflection of the performance of the recognizer, because with a sufficient number of insertions the score can be made 100%. A better procedure is to express quantities as a percentage of $(R+I)$. Thus, we may define:

$$\% \text{ correct} = 100 \times C/(R+I) \tag{11a}$$

$$\% \text{ insert} = 100 \times I/(R+I) \tag{11b}$$

$$\% \text{ delet} = 100 \times D/(R+I). \tag{11c}$$

The top rows on the right-hand side of Tables II, III, IV, respectively, present results according to these equations.

For comparison, we ran a modified version of the "phonetic recognizer" of Levinson (1986) using the same database, KBDY. In the modified version, the basic unit is a phoneme (47 phonemes, shown in Table I, were used); each phoneme was represented by 47 conditional probability densities for the observations (taking left context into account), and 47 conditional probability densities for the durations. The observation vectors were the cepstra plus delta–cepstra and energy plus delta–energy (with a window of five frames). The analysis conditions were as mentioned in Section 2.1: an LPC-based cepstrum analysis was performed on each sentence in the database. The LPC order was 12; the order of the cepstrum vectors obtained from these LPC vectors was 11. The window length was 30 ms, and the overlap between successive windows was 10 ms. Energy was averaged over a 30 ms window.

The results of this experiment are shown on the bottom rows of Tables II, III, IV, computed according to Equations (10) and (11), respectively. A comparison of corresponding entries in the top and bottom rows of these tables shows that with either measure of performance, our recognizer gives significantly more correct phonemes and significantly fewer insertions than the recognizer of Levinson (1986).

3.2. Detailed experiment

In this section we conducted a series of experiments aimed at studying how the performance of our recognizer depends on the statistical complexity of the state models. The performance of the recognizer was measured in the manner described in Section 3.1.

We decided to consider only those diphones for which we found at least three tokens in the database. For the KBB database, we obtained such ensembles for 948 different diphones. The number of tokens in any such ensemble is between 3 and 79, with an average of 37. For each of these ensembles we derived a template in the manner described in Section 2.2. The observation vectors, which were derived in the manner described in Section 2.1, included energy, which was averaged over a 10 ms window.

We derived five different non-stationary HMMs based on these states. The models differ in the assumptions concerning the covariance matrix $\mathbf{\Phi}$ for each state. As before we assume $\mathbf{\Phi}$ to be block diagonal, so each model is completely specified by the diagonal blocks $\mathbf{\Phi}_{ii}$. We have considered the following types of matrices:

(a) In the first model we assumed $\Phi_{ii} = \sigma^2 \mathbf{I}$, where $\mathbf{I}$ is the identity matrix. We used the same value of σ^2 for all the states, and for all the diagonal blocks. This value (about 0·04) was the average variance of a cepstral coefficient over all the data.

(b) In the second model we assumed Φ_{ii} to be a diagonal matrix. The matrix was assumed to be constant for each phoneme of the diphone. These matrices were prototype matrices, derived by pooling data from structurally similar diphones, as described in the preliminary experiment. Thus, eight different prototype diagonal matrices were estimated—two for vowel–vowel diphones, two for vowel–consonant diphones, etc.

(c) In the third model we improved on model (b) by estimating *state-dependent* diagonal matrices for each of 293 states. For these states, we estimated two state-dependent matrices, one for the first phoneme and one for the second phoneme of the diphone. For the rest of the states, as in the preliminary experiment, we used prototype matrices.

(d) This model was the same as model (b), except the eight prototype matrices were *full* $p \times p$ matrices.

(e) Finally, we improved on model (d) by estimating *state-dependent* full covariance matrices for each of 293 states. For the rest of the states, as in the preliminary experiment, we used prototype matrices.

Each of these models was used to decode a set of 50 sentences outside the training set, using exactly the same procedure as outlined in Section 3.1. The 50 test sentences are the Harvard PB (phonetically balanced) sentence list. The hand-labeled transcription of these test sentences had 1225 phonemes, and the phonemes "schwa" and "R" appeared a total of 161 times.

Tables V, VI and VII give the results of these experiments. Comparison of the corresponding entries in these tables shows that the first four assumptions about the covariance matrices lead to essentially the same performance. The full covariance state-dependent matrices give the best results. Derivation of such matrices for all the states should improve the recognition accuracy.

3.3. Equivalence classes

Some phonemes, in certain contexts, have such similar spectra that one might consider a recognition scheme in which the phoneme decoder identifies only the "confusable group", or equivalence class, to which a phoneme belongs. Subsequent processing could disambiguate the recognized phoneme string, on the basis of context, or grammatical constraints, etc. We believe that this might be a good strategy, especially for fluent speech.

Hypothesizing such a recognizer, it is of interest to evaluate the error rate of our transcriptions on the assumption that a recognized phoneme is correct if its equivalence class is correctly identified.

Table VIII was compiled for us by J. P. Olive, and gives a list of phonemes that are exchangeable in the sense described above. Based on this table, we modified the alignment procedure mentioned in Section 3.1 to score the correctness of equivalence classes rather than phonemes. The results of such an alignment are shown in Tables IX and X. Note that since "schwa" and "R" are equivalent to other phonemes, we do not treat them differently from other phonemes in this alignment procedure.

TABLE V. New HMM with different statistical models, "R" and "schwa" assumed wrong, database KBB

	Equation (10)			Equation (11)		
	Per cent correct	Per cent insert	Per cent delet	Per cent correct	Per cent insert	Per cent delet
Diag. Φ, diag. Φ_{ii}, $\sigma_i^2 = const.$	56·3	12·6	4·2	50·0	11·2	3·7
Diag. Φ, prototype diag. Φ_{ii}	56·0	12·1	4·0	50·0	10·8	3·6
Diag. Φ, state-dependent diag. Φ_{ii}	56·1	12·2	3·5	50·0	10·8	3·1
Diag. Φ, prototype full Φ_{ii}	57·6	11·1	4·2	51·8	10·0	3·7
Diag. Φ, state-dependent full Φ_{ii}	61·1	11·2	3·8	54·9	10·1	3·4

TABLE VI. New HMM with different statistical models, "R" and "schwa" ignored, database KBB

	Equation (10)			Equation (11)		
	Per cent correct	Per cent insert	Per cent delet	Per cent correct	Per cent insert	Per cent delet
Diag. Φ, diag. Φ_{ii}, $\sigma_i^2 = const.$	64·8	14·5	4·8	56·7	12·6	4·2
Diag. Φ, prototype diag. Φ_{ii}	64·5	13·9	4·6	56·6	12·2	4·0
Diag. Φ, state-dependent diag. Φ_{ii}	64·6	14·0	4·0	56·6	12·3	3·5
Diag. Φ, prototype full Φ_{ii}	66·3	12·8	4·8	58·7	11·3	4·3
Diag. Φ, state-dependent full Φ_{ii}	70·3	12·9	4·3	62·3	11·4	3·8

TABLE VII. New HMM with different statistical models, "R" and "schwa" assumed correct, database KBB

	Equation (10)			Equation (11)		
	Per cent correct	Per cent insert	Per cent delet	Per cent correct	Per cent insert	Per cent delet
Diag. Φ, diag. Φ_{ii}, $\sigma_i^2 = const.$	69·5	12·6	4·2	61·7	11·2	3·7
Diag. Φ, prototype diag. Φ_{ii}	69·1	12·1	4·0	61·7	10·8	3·6
Diag. Φ, state-dependent diag. Φ_{ii}	69·2	12·2	3·5	61·7	10·8	3·1
Diag. Φ, prototype full Φ_{ii}	70·7	11·1	4·2	63·6	10·0	3·7
Diag. Φ, state-dependent full Φ_{ii}	74·2	11·2	3·8	66·7	10·1	3·4

TABLE VIII. Equivalence classes for phonemes

Phoneme	Equivalents	Phoneme	Equivalents
ɛ	æ, I, e^y, ə	h	
ɔ	o^w, ɑ, ʌ	s	t
ɑ	a^w, ʌ, $ɑ^y$	ʃ	č
u	ʊ, w	z	s
ɚ	r	ʒ	ʃ
$ɑ^y$	ɑ, I, ʌ	č	ʃ
e^y	I, i	ǰ	ʒ
a^w	ɑ, ʌ, u, ʊ	ɵ	f, ð
ə	I, ɛ, ʊ, ʌ	ð	ɵ
I	ɛ, e^y, ə	f	ɵ, v, p
æ	ɛ	v	b, w, f
ʌ	ɑ, a^w, $ɑ^y$, ə	m	v, b, n
ʊ	u, a^w, ə	n	ŋ, d, ð, m
$ɔ^y$	ɔ, I	ŋ	n, g
i	y	p	f, v, b
o^w	ɔ, u, ʊ	t	s, d
l	w	k	g
r	ɚ	b	v, w, p, m
w	l, ʊ, u, a^w, o^w	d	t, n
y	i	g	ŋ, k

As expected, the error rates are substantially lower than those for the strict alignment. However, the ordering of the entries remains unchanged.

4. Future directions

One of our major future goals is to remove the (temporary) restriction of the covariance matrix $\mathbf{\Phi}$ to block diagonal. As discussed in the Introduction, a full covariance matrix would completely account for the statistical dependence on each other of all the observations in a state.

Another objective is to make the recognizer speaker-independent. For such a task, the state would undoubtedly have to be represented by more than one template.

Both these objectives require much more data than we have at present. The extension to a full covariance matrix might still be possible by additional labeling of data for one

TABLE IX. New HMM vs. base-line HMM, using a modified Levinshtein measure, database KBDY

	Equation (10)			Equation (11)		
	Per cent correct	Per cent insert	Per cent delet	Per cent correct	Per cent insert	Per cent delet
New HMM	73·9	21·2	4·0	61·0	17·5	3·7
Base-line HMM	70·5	49·2	3·5	47·2	33·0	2·3

TABLE X. New HMM with different statistical models, modified Levinshtein measure, database KBB

	Equation (10)			Equation (11)		
	Per cent correct	Per cent insert	Per cent delet	Per cent correct	Per cent insert	Per cent delet
Diag. Φ, diag. Φ_{ii}, $\sigma_i^2 = const.$	76·5	12·6	6·0	67·9	11·2	5·3
Diag. Φ, prototype diag. Φ_{ii}	74·7	12·1	5·3	66·6	10·8	4·7
Diag. Φ, state-dependent diag. Φ_{ii}	75·1	12·2	4·5	67·0	10·8	4·0
Diag. Φ, prototype full Φ_{ii}	74·4	11·1	5·2	67·0	10·0	4·7
Diag. Φ, state-dependent full Φ_{ii}	78·1	11·2	4·7	70·3	10·1	4·2

speaker. However, it is much more desirable to have some automatic (or semi-automatic) means to collect the data. For this, we can think of several possibilities. One, for example, is to use the states derived from hand-segmented data to segment data with a known transcription automatically. This can be iterated to improve the statistical representation of our states (Lee *et al.*, 1990). Other possibilities are extensions of automatic segmentation techniques used for deriving "phoneme-like" units, e.g. the one given in Lee *et al.* (1990).

We would like to express our sincere thanks to J. P. Olive, for not only providing us with his labeled databases, but also for many discussions and suggestions during the course of this work. We would also like to thank Roberto Pieraccini for supplying us with a program for aligning symbol sequences. A slightly modified version of his program was used in the evaluation of transcriptions described in Section 3. Thanks also to Andrej Ljolje for providing us with a program for the left-context dependent version of Levinson (1986). A slightly modified version of his program was used to measure the performance of the base-line system.

References

Bahl, L. R., Jelinek, F. & Mercer, R. L. (1983). A maximum likelihood approach to continuous speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **PAMI 5**, 179–190.

Deng, L. (1992). A generalized hidden Markov model with state-conditioned trend functions of time for the speech signal. To appear in Signal Processing.

Lee, C. H., Rabiner, L. R., Pieraccini, R. & Wilpon, J. G. (1990). Acoustic modeling for large vocabulary speech recognition. *Computer Speech and Language*, **4**, 127–165.

Levinson, S. E. (1986). Continuously variable duration hidden Markov models for speech recognition. *Computer Speech and Language*, **1**, 29–46.

Levinson, S. E. (1987). Continuous speech recognition by means of acoustic/phonetic classification obtained from a hidden Markov model. In *Proceedings of ICASSP '87*, Dallas, Texas, April 6–9, 1987, pp. 93–96.

Ostendorf, M. O. & Roukos, S. (1989). A stochastic segment model for phoneme-based continuous speech recognition. *IEEE Transactions on ASSP*, **ASSP 37**, 1857–1869.

Roucos, S. & Dunham, M. O. (1987). A stochastic segment model for phoneme-based continuous speech recognition. In *Proceedings of ICASSP '87*, Dallas, Texas, April 6–9, 1987, pp. 73–76.

Wellekens, C. J. (1987). Explicit time correlation in hidden Markov models for speech recognition. In *Proceedings of ICASSP '87*, Dallas, Texas, April 6–9, 1987, pp. 384–386.

Wilpon, J. G. & Rabiner, L. R. (1985). A modified *k*-means clustering algorithm for use in speaker-independent isolated word recognition. *IEEE Transactions on ASSP*, **ASSP 33**, 587–594.