Vector Quantization Bigram Hidden Markov Modelling For Improved Phoneme Recognition

G. Wong and S. J. Young
Cambridge University Engineering Department,
Trumpington Street, Cambridge, CB2 1PZ, England
email: gw@uk.ac.cam.eng

July 21, 1992

Abstract

The development of accurate and robust phonetic models is essential for high-performance continuous speech recognition since the words themselves are mapped out as a sequence of phonemes. One approach is to model the time dependencies of the acoustic features in a phoneme more accurately. Short-time correlation between successive feature vectors (condensed as vector quantization codes) is modelled as discrete emission probabilities embedded in the observation process of a Hidden Markov Model (HMM). Reestimation equations in an Expectation-Maximization (EM) framework are presented for the training of such a model, as well as the Viterbi decoding algorithm necessary for phoneme based continuous speech recognition. The Expectation step in the parameter reestimation stage calculates the log likelihood of the observation sequence and the Maximization step yields the estimates of the state transition terms and conditional output pdf parameters separately. A Lagrange interpretation of the derived reestimation formulas is also presented. Recognition results using the TIMIT database are compared with conventional discrete Hidden Markov modelling methods and a measurable improvement (14% error rate reduction) has been achieved. Implementation and several aspects of this modelling method are discussed with possible extensions for further improvements.

1 Introduction

One of the problems generally encountered in current HMM systems is that the correlation between adjacent observation frames is inaccurately modelled because each VQ label is treated as if it had no dependency upon other labels. However, a look at the VQ pattern of sentences, specificially for the mel-frequency scale cepstral coefficients (mfcc), indicates that for each VQ index that has a high probability of occurence, there are several other VQ indices that are also likely to have high probabilities. Thus, successive feature vectors, or their vector quantization counterparts, can be strongly correlated.

Information about co-occuring VQ codes has been implicitly used to smooth conventional HMM parameters [1, 2]. The idea behind this technique is to use this information to differentiate between likely unobserved VQ symbols from unlikely ones. In the stochastic segment model [3] and C. H. Lee's work [4], each speech unit is modelled as a sequence of acoustic segments, within which a high degree of correlation exists among the frames. This is precisely what a standard HMM tends to model by associating frames within a fairly stationary region to a particular HMM state. Correlation over a long time span has been effectively modelled by the Markov chain in Kriouile's work where the usual first-order Markov chain is replaced by a second-order one [5].

Because of the strong correlation among observations assigned to the same state, an attempt has been made to extend the modelling of the correlation by the observation process by explicitly considering the probability of a VQ symbol given that a previous one has occurred. There is also some empirical evidence that the independence of frames assumption in the output probability distribution in an HMM is inaccurate; by computing the probability that a sequence of VQ codes is generated given the HMM and the probability of that same sequence, but given a maximum likelihood estimate of a modified model modelling explicitly frame correlations, it is found that the latter probability is generally significantly higher. These results have been observed in both training and recognition for those conditionally determined models.

Recognition results are invariably improved when supplemented by time-derivative information e.g. [6]. Because of the way in which both current VQ information is carried in the output process as well as the previous VQ symbol, a measure of differential information is implicitly present in the modified modelling. Whereas this information can be considered to spread over the time span of one frame separation, the differential mfcc preprocessing normally done for phone recognition tasks covers a time duration of four frames (in the current implementation).

VQ bigram information in a more static context has been applied to multi-section codebook [7], isolated word recognition [8, 9], speaker adaptation [10] and even noisy speech recognition [11]. Explicit short-time correlation modelling by the output probability process has been attempted by Brown [12] for isolated word recognition. An error rate reduction of 26% was obtained in a 2000-word vocabulary, speaker-dependent isolated word experiment. No reestimation equations are given either for the training and testing phase in his thesis. Although those equations can be arrived at intuitively, the dimension of the problem for large speech tasks requires careful attention to its implementation when conducting experiments. The conditional acoustic labels are mapped to a smaller number of sub-classes by some merging procedure to maximize the average information between the class of a VQ index and the next VQ index. A correlated continuous density output probability density function (pdf) has also been suggested by Brown [12] and Wellekens [13] where the way in which the feature vector at time $t-1$ differs from the mean of its output distribution will be correlated to the way in which the feature vector at time t differs from the mean of the output distribution from which it is generated. However, no experimental validation has been reported.

Accurate phoneme recognition is an intermediate but highly essential task for large vocabulary continuous speech recognition since the words themselves are mapped out as a sequence of phonemes. More specifically, large scale word recognition tasks entail word HMMs being built from a concatenation of the phoneme HMMs. This work is concerned with refining the output process probability mechanism of an HMM from a zero-order to a first-order Markov process in an attempt to improve recognition performance on the TIMIT phoneme database. Another approach for better and more accurate in-class modelling has been modelling context dependency of a particular phoneme based upon the immediate surrounding phonemes. Considerable improvements have been achieved this way [1] but the complexity of the system has now increased in the implementation, computation, and memory requirements. One of the objectives of this work is to keep in check the number of free parameters to be estimated and yet try to achieve high levels of recognition performance. Indeed, Robinson [14] has shown that using neural networks the number of parameters are about an order of magnitude lower than for conventional HMMs for similar performance levels.

The organization of this report is as follows. Section 2 briefly reviews the Expectation-Maximization algorithm for solving Maximum Likelihood problems. Section 3 starts by considering the assumptions necessary for estimation of the standard HMM and the relaxation in the output process probability assumption. The EM algorithm for standard HMM estimation is reviewed and the extension to the more detailed conditional HMMs is considered, as well as for coping with multiple codebooks. The reestimation equations are presented in section 4 with implementation details, and throughout the contrast with standard HMMs is stressed. An alternative derivation of the reestimation equations is demonstrated using Lagrangian techniques. The corresponding modifications to the Viterbi decoding equations using the conditional HMMs are presented in section 5. Empirical continuous phoneme recognition results for the DARPA TIMIT task are then contrasted with baseline experiments and an attempt at crude variable frame rate analysis of the VQ observation sequences is described in section 6. Finally, the last section 7 discusses the implications of the results, general conclusions using the conditional models and possible extensions of the technique.

2 EM theory

The principle of Expectation-Maximization (EM) theory is briefly outlined here. The main general references for this section are by Dempster and Redner [15, 16] The ones more specific to HMMs are by Baum [17, 18] and by Liporace [19].

The EM algorithm is a general approach for maximizing a likelihood or posterior (Bayesian) function when some of the data are 'missing' in some sense, and observation of that missing data would greatly simplify the estimation of parameters. Without that missing data component introduced in the likelihood function, the likelihood function of the original data may be too difficult or impractical to maximize or simply not accurate enough for some time sequential problems. In the present HMM case which models sequences of measurement vectors, data are 'missing' not because of any censoring or misrecording, but because of the superposition of the state sequence whose behaviour is governed by a first-order Markov chain. This means that the more basic data – the state from which the observation is emitted at time t cannot be observed.

Let Y be the observed 'incomplete data' which has the pdf $P(Y|\theta)$, from which we wish to estimate the parameter vector θ. The maximum likelihood (ML) estimator of θ based on the available incomplete data, is given by

$$\hat{\theta}_{ml,Y} = \arg\{\max_{\theta} L_Y(\theta)\} = \arg\{\max_{\theta} \log P(Y|\theta)\}, \tag{1}$$

where $L_Y(\theta)$ is the likelihood function of the incomplete data.

At this point, it is assumed that the complete data X have been chosen in such a way that computing the ML estimator of θ from the complete data, i.e. solving

$$\hat{\theta}_{ml,X} = \arg\{\max_{\theta} L_X(\theta)\} = \arg\{\max_{\theta} \log P(X|\theta)\} \tag{2}$$

is significantly simpler than solving (1). $L_X(\theta)$ is the likelihood function of the complete data.

The incomplete data is related to the complete data X through a non-invertible many-to-one transformation:

$$Y = J(X). \tag{3}$$

The transformation $J(.)$ relating X to Y can be any non-invertible transformation. There may be many possible complete data specifications that will generate the observed data – the EM algorithm can therefore be implemented in many possible ways. The formulation of the complete data is crucial because a good one will reduce the complexity and convergence time of the algorithm. The pdf of X, which is also indexed by θ, is related to Y as follows:

$$P(Y|\theta) = \int_{X(Y)} P(X|\theta)\, dX. \tag{4}$$

The probability distribution of the parameter vector θ conditioned on the data vector Y using Bayes rule is:

$$P(\theta|Y) = \frac{P(Y|\theta)P(\theta)}{P(Y)}. \tag{5}$$

For Bayesian estimation, some *a priori* information is incorporated in the estimate by specification of the prior $P(\theta)$. Since $P(Y)$ is a constant for a given data vector belonging to a particular class, Maximum *a posteriori* (MAP) estimation yields the following statement:

$$\max_{\theta} B(\theta|Y) = \log P(Y|\theta) + \log P(\theta). \tag{6}$$

The term $B(\theta|Y)$ is known as the Bayesian or log posterior function.

It then follows that due to this many-to-one X to Y mapping

$$P(X|Y;\theta) = \frac{P(X,Y|\theta)}{P(Y|\theta)} = \frac{P(X|\theta)I_Y(X)}{P(Y|\theta)}, \tag{7}$$

where $I_Y(X)$ is the indicator function which is equal to 1 if X results in Y and equal to 0 otherwise. For an HMM, this indicator is the state from which the observation is emitted.

Consider the denominator term of (7). For any $\hat{\theta}^n$, the parameter vector at the n-th iteration in the reestimation algorithm, the following relationship applies:

$$E_X\Big\{\log P(Y|\theta)|Y;\hat{\theta}^n\Big\} = \int_{X(Y)} \log P(Y|\theta)P(X|Y;\hat{\theta}^n)dX = \log P(Y|\theta), \tag{8}$$

where $E_X\{.\}$ is the expectation operator with respect to X, the complete data. $P(X|Y;\hat{\theta}^n)$ is the conditional pdf of the complete data, given the incomplete data and the estimate $\hat{\theta}^n$.

Combining (7) and (8) by taking log and then the expectation operator gives

$$\log P(Y|\theta) = E_X\Big\{\log P(X|\theta)|Y;\hat{\theta}^n\Big\} - E_X\Big\{\log P(X|Y,\theta)|Y;\hat{\theta}^n\Big\}. \tag{9}$$

Substituting (9) into (6) results in

$$B(\theta|Y) = Q(\theta|\hat{\theta}^n) - E_X\Big\{\log P(X|Y,\theta)|Y;\hat{\theta}^n\Big\}, \tag{10}$$

where

$$Q(\theta|\hat{\theta}^n) = E_X\Big\{\log P(X|\theta)|Y;\hat{\theta}^n\Big\} + \log P(\theta). \tag{11}$$

The first term is the conditional expectation of the log likelihood of the complete data, given the observed incomplete data Y and $\hat{\theta}^n$.

So, (11) can be re-written as:

$$Q(\theta|\hat{\theta}^n) = L(\theta|\hat{\theta}^n) + \log P(\theta), \tag{12}$$

where

$$L(\theta|\hat{\theta}^n) = E_X\Big\{\log P(X|\theta)|Y;\hat{\theta}^n\Big\} \tag{13}$$

From Jensen's inequality [20] (pages 49-50) for any $\hat{\theta}^{n+1} \neq \hat{\theta}^n$,

$$E_X\Big\{\log P(X|Y,\hat{\theta}^{n+1})|Y;\hat{\theta}^n\Big\} \leq E_X\Big\{\log P(X|Y,\hat{\theta}^n)|Y;\hat{\theta}^n\Big\}, \tag{14}$$

with equality if and only if

$$\log P(X|Y;\hat{\theta}^{n+1}) = \log P(X|Y;\hat{\theta}^n). \tag{15}$$

It follows that a sufficient condition for $B(\hat{\theta}^{n+1}|Y) > B(\hat{\theta}^n|Y)$ is
$Q(\hat{\theta}^{n+1}|\hat{\theta}^n) > Q(\hat{\theta}^n|\hat{\theta}^n)$ since the second term of (10) is guaranteed not to decrease by Jensen's inequality.

In general, if $B(\theta|Y)$ is not unimodal, the EM approach at best assures convergence of the sequence $\{\theta^{\mathbf{n}}\}$ to a stationary value. The convergence point will normally not be the global maximum of the object function for a complicated problem – several starting points in the initial parameter vector space may be needed to locate the best maxima. For the finer points of the convergence of the EM algorithm, refer to Wu's work [21].

The EM algorithm is the application of those two steps below, in an iterative way, until a pre-defined threshold is attained.

The E-step

Starting with an estimate of the parameter vector $\hat{\theta}^n$ for the n-th iteration of the reestimation algorithm, the EM algorithm for MAP estimation consists of the Expectation step (E-step) whereby

$$E_X\Big\{\log P(X|Y,\theta)|Y;\hat{\theta}^n\Big\} \tag{16}$$

is formed i.e. the expected value of the logarithm of the pdf of the complete data is evaluated, where the expectation is with respect to the probability measure defined by the incomplete data and the current parameters. The E-step thus finds the conditional expectation of the sufficient statistic for the complete data log likelihood. Note that the E-step is not affected by the prior term.

The M-step

The Maximization step (M-step) corresponds to the maximization of the log likelihood function, $L(\theta|\hat{\theta}^n)$ with respect to θ. This leads to a new parameter estimate $\hat{\theta}^{(n+1)}$:

$$\hat{\theta}^{(n+1)} = \arg\{\max_\theta L(\theta|\hat{\theta}^{(n)})\}. \tag{17}$$

For the posterior function, solve

$$\arg\{\max_\theta Q(\theta|\hat{\theta}^n)\}, \tag{18}$$

where

$$Q(\theta|\hat{\theta}^n) = E_X\Big\{\log P(X|\theta)|Y;\hat{\theta}^n\Big\} + \log P(\theta). \tag{19}$$

The complete data log likelihood is maximized with respect to the unknown parameters, with the conditional expectation of the sufficient statistics substituted in place of their unknown values. The choice of the prior function will affect this maximization step of the EM algorithm for the posterior function.

For the log likelihood function where $L_X(\theta)$ is defined on the true complete data, $L(\theta|\hat{\theta}^{(n)})$ uses the conditional expectation of the complete data. The maximization of $L(\theta|\hat{\theta}^{(n)})$ with respect to θ is therefore of the same complexity as the maximization of $L_X(\theta)$. Because of this, the EM algorithm is an attractive alternative to the direct evaluation of (1) only if the solution to (2) can be computed relatively easily. Solving the more general Q function with the prior term depends on a judicious choice of this prior term. If components of the complete data X are independent, the complete data likelihood function is a linear function of the incomplete data – the M-step only requires the optimization of a set of those functions. This decoupling is fundamental to HMM problems, as will be seen in the subsequent sections.

3 Reestimation of parameters

This section starts by reviewing the assumptions made in the formulation of the standard HMM. A relaxation in the assumption about the output process probability is then made. The EM algorithm is reviewed in its application to standard HMMs. A simple modification is then introduced to cope with the more detailed models, multiple codebook observation sequence and multiple observation sequences.

3.1 Conventional HMM formulation and extended formulation

In standard Hidden Markov modelling, the observation probability process is defined only from the current HMM state and is independent of both the preceding states and of the precedingly emitted observation vectors. This crude assumption allows computation of parameters associated with the HMM parameters to a manageable level. The decoupling of the observable observation terms and the unobservable state terms is the key to the Hidden Markov modelling problem. A definition of terms for standard HMMs is given below.

s_t	state at time t
N	number of states in Markov chain
M	number of VQ prototype spectra
T	length of observation sequence
$A = [a_{ij}]$	$P(s_t = j \vert s_{t-1} = i)$
$B = [b_{jk}]$	$P(O_t = k \vert s_t = j)$
$\pi = [\pi_i]$	$P(s_0 = i)$
$aF = [aF_i]$	$P(s_T = i)$
$\theta \equiv (\pi, A, aF, B)$	the complete parameter set
$\mathbf{O} = [O_t]$	the observation sequence

For a particular state sequence s,

$$P(s|\theta) = \pi_{s_0}\Big(\prod_{t=1}^{T} a_{s_{t-1}s_t}\Big) aF_{s_T}, \tag{20}$$

and

$$P(\mathbf{O}|s,\theta) = \prod_{t=1}^{T} b_{s_t}(O_t). \tag{21}$$

Summing over all possible state sequences,

$$\begin{aligned} P(\mathbf{O}|\theta) &= \sum_{s} P(s,\mathbf{O}|\theta) && \text{(22a)} \\ &= \sum_{s} P(s|\theta)P(\mathbf{O}|s,\theta) && \text{(22b)} \end{aligned}$$

or

$$P(\mathbf{O}|\theta) = \sum_{s} \pi_{s_0} \Big(\prod_{t=1}^{T} a_{s_{t-1}s_t} b_{s_t}(O_t) \Big) aF_{s_T}. \tag{23}$$

The objective of the reestimation algorithm is to maximize log $P(\mathbf{O}|\theta)$.

The output independence assumption can be relaxed by making the probability of the whole observation sequence dependent on the state transition taken at time t and on the observation frame at the previous time frame $t-1$. Thus, (21) is replaced by:

$$P(\mathbf{O}|s,\theta) = \prod_{t=1}^{T} b_{s_t}(O_t|O_{t-1}). \tag{24}$$

Therefore, (23) now becomes:

$$P(\mathbf{O}|\theta) = \sum_{s} \pi_{s_1} \Big(\prod_{t=1}^{T} a_{s_{t-1}s_t} b_{s_t}(O_t|O_{t-1}) \Big) aF_{s_T}. \tag{25}$$

The standard HMM has therefore been made less restrictive by defining a *correlated observation process probability.* The resulting HMM structure will be dubbed a *vector quantized bigram HMM* (VQBHMM). The VQ bigram HMM state and output probability process is illustrated schematically in figure 1. At time t, the process is assumed to be in state s_t and a conditional output VQ pair $[O_t|O_{t-1}]$ is generated according to some state-dependent output pdf $b_{s_t}(.)$. The output probability elements of this VQBHMM will be indexed in the following way:

$$b_{jkm} = b_j(k|m) = P(O_t = k|s_t = j, O_{t-1} = m). \tag{26}$$

Thus, the parameters of the VQBHMM are the initial probabilities π_i, final probabilities aF_i, the first-order Markov transition probabilities a_{ij} and the first-order Markov output distributions b_{jkm}.

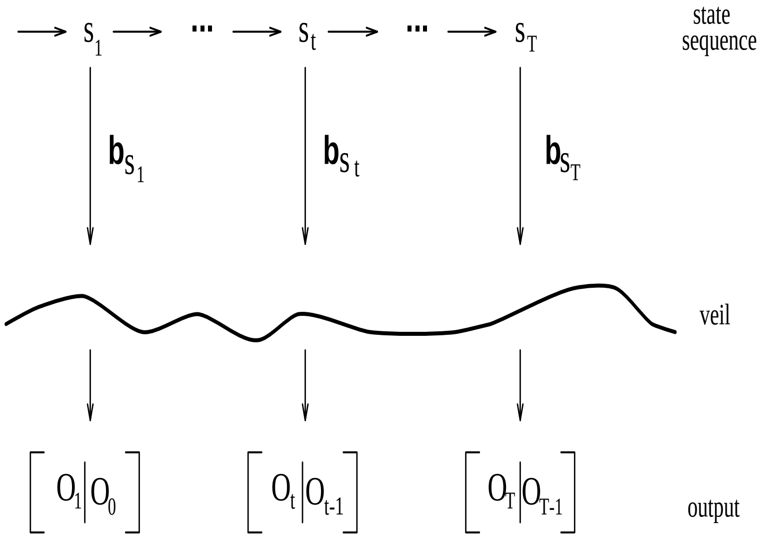

Figure 1: Generating a state sequence and conditional observation sequence from a VQ bigram HMM.

3.2 EM formulation for conventional HMM

The *incomplete* but *observable* data Y is $\mathbf{O}$. The *complete* data X are the set of signals $\{\mathbf{O}, s\}$ where s is a particular (unobservable) state sequence. The parameters to be estimated θ are the set (π, A, aF, B).

The E-step

From (13) and (16), the E-step involves the calculation of the log likelihood:

$$
\begin{aligned}
L(\theta|\hat{\theta}^n) &= E_X\Big\{\log P(X|\theta)|Y;\hat{\theta}^n\Big\} && \text{(27a)}\\
&= E_{\mathbf{O},s}\Big\{\log P(s,\mathbf{O}|\theta)|\mathbf{O};\hat{\theta}^n\Big\} && \text{(27b)}\\
&= \Big\{\log P(\mathbf{O}|\theta)|\mathbf{O};\hat{\theta}^n\Big\}. && \text{(27c)}
\end{aligned}
$$

The M-step

From the above equation (27), (17) and (23), the M-step involves:

$$
\begin{aligned}
\hat{\theta}^{(n+1)} &= \arg\Big\{\max_{\theta} L(\theta|\hat{\theta}^{(n)})\Big\} && \text{(28a)}\\
&= \arg\Big\{\max_{\theta} E_{\mathbf{O},s}\Big(\log \pi^n_{s_0} + \sum_{t=1}^{T}\log a^n_{s_{t-1}s_t} + \log aF^n_{s_T} + \sum_{t=1}^{T}\log b^n_{s_t}(O_t)\Big)\Big\} && \text{(28b)}\\
&= \arg\Big\{\max_{\pi_i} E_{\mathbf{O},s}\Big(\log \pi^n_{s_0}\Big)\Big\} + \arg\Big\{\max_{a_{ij}} E_{\mathbf{O},s}\Big(\sum_{t=1}^{T}\log a^n_{s_{t-1}s_t}\Big)\Big\} + && \text{(28c)}\\
&\quad \arg\Big\{\max_{aF_i} E_{\mathbf{O},s}\Big(\log aF^n_{s_T}\Big)\Big\} + \arg\Big\{\max_{b_{jk}} E_{\mathbf{O},s}\Big(\sum_{t=1}^{T}\log b^n_{s_t}(O_t)\Big)\Big\}. && \text{(28d)}
\end{aligned}
$$

A modified $L(\theta|\hat{\theta}^{(n)})$ simplifies the estimation of each of those components. For reestimation, an auxiliary function, Q_1, of the current model θ and the n-th iteratively estimated model $\theta^{\mathbf{n}}$ is defined. This auxiliary function:

$$Q_1 = Q(\theta, \theta^{\mathbf{n}}) = \sum_s P(s, \mathbf{O}|\theta) \log P(s, \mathbf{O}|\theta^{\mathbf{n}}) \tag{29}$$

has been surmised by Baum [18]. Similarly, define

$$Q_2 = Q(\theta, \theta) = \sum_s P(s, \mathbf{O}|\theta) \log P(s, \mathbf{O}|\theta). \tag{30}$$

Then,

$$\begin{aligned} Q_1 - Q_2 &= \sum_s P(s, \mathbf{O}|\theta) \log \frac{P(s, \mathbf{O}|\theta^{\mathbf{n}})}{P(s, \mathbf{O}|\theta)} && \text{(31a)} \\ &\leq \sum_s P(s, \mathbf{O}|\theta) \Big\{ \frac{P(s, \mathbf{O}|\theta^{\mathbf{n}})}{P(s, \mathbf{O}|\theta)} - 1 \Big\} && \text{(31b)} \\ &= P(\mathbf{O}|\theta^{\mathbf{n}}) - P(\mathbf{O}|\theta). && \text{(31c)} \end{aligned}$$

The inequality arises because the function $y = \log x$ is bounded above by any tangent line [19]. In fact, the tangent line at $x = 1$ is the line $y = x - 1$, so that $\log x \leq x - 1$, with equality if and only if $x = 1$. If $Q_1 > Q_2$, then $P(\mathbf{O}|\theta^{\mathbf{n}}) > P(\mathbf{O}|\theta)$. If a method of reestimating parameter vector θ^n can be found that makes (31) positive, then the likelihood is maximized in the process or at least not made worse.

Applying the logarithm operator to $P(s, \mathbf{O}|\theta)$ (from (22) and (23)),

$$\begin{aligned} \log P(s, \mathbf{O}|\theta^{\mathbf{n}}) &= \sum_i \delta(s_0, i) \log \pi^n_{s_0} + && \text{(32a)} \\ & \sum_{t=1}^{T} \sum_{i,j} \delta(s_{t-1}, i)\delta(s_t, j) \log a^n_{s_{t-1}s_t} + && \text{(32b)} \\ & \sum_{t=1}^{T} \sum_{j,k} \delta(s_t, j)\delta(O_t, k) \log b^n_{s_t}(O_t) + && \text{(32c)} \\ & \sum_i \delta(s_T, i) \log aF^n_{s_T} && \text{(32d)} \end{aligned}$$

where δ is the Dirac delta function.

Hence, by substituting the above (32) into (29), and because the state transition terms can be decoupled from the output process terms (following the same notation as put forward by Levinson [22]):

$$\begin{aligned} Q_1 &= \sum_{i=1}^{N} \sum_{j=1}^{N} c_{ij} \log a^n_{ij} + && \text{(33a)} \\ & \sum_{j=1}^{N} \sum_{k=1}^{M} d_{jk} \log b^n_j(k) + && \text{(33b)} \\ & \sum_{i=1}^{N} e_i \log \pi^n_i + && \text{(33c)} \\ & \sum_{i=1}^{N} z_i \log aF^n_i && \text{(33d)} \end{aligned}$$

where

$$c_{ij} = \sum_s P(s, \mathbf{O}|\theta) n_{ij}(s), \tag{34}$$

$$d_{jk} = \sum_s P(s, \mathbf{O}|\theta) m_{jk}(s), \tag{35}$$

$$e_i = \sum_s P(s, \mathbf{O}|\theta) r_i(s), \tag{36}$$

and

$$z_i = \sum_s P(s, \mathbf{O}|\theta) h_i(s). \tag{37}$$

$n_{ij}(s)$ is the number of transitions from state i to state j for the s-th state sequence, $m_{jk}(s)$ is the number of times symbol k is generated from state j for the s-th state sequence, $r_i(s)$ is 1 if initial state is i, and 0 otherwise, and finally, $h_i(s)$ is 1 if the final state is i, and 0 otherwise.

Invoking the following lemma from Levinson's paper [22] (page 1042)

lemma 1 *: If $c_i > 0, i = 1, \cdots, N$, then subject to the constraint $\sum_i x_i = 1$, the function*

$$F(\mathbf{x}) = \sum_i c_i \log x_i \tag{38}$$

attains its unique global maximum when

$$x_i = \frac{c_i}{\sum_i c_i}. \tag{39}$$

(33) yields a sum of independent expressions of the type described by the above lemma and so the new parameters can be reestimated as shown in the following equations. In the same step, $Q_1 = Q(\theta, \theta^{\mathbf{n}})$ is maximized. The n-th iterative estimates are thus given by:

$$a^n_{ij} = \frac{c_{ij}}{\sum_j c_{ij}} \tag{40a}$$

$$= \frac{P(s_t = j | s_{t-1} = i, \mathbf{O}, \theta)}{P(s_t = i|\mathbf{O}, \theta)}, \tag{40b}$$

$$b^n_j(k) = \frac{d_{jk}}{\sum_k d_{jk}} \tag{41a}$$

$$= \frac{P(O_t = k | s_t = j, \mathbf{O}, \theta)}{P(s_t = j|\mathbf{O}, \theta)}, \tag{41b}$$

$$\pi^n_i = \frac{e_i}{\sum_i e_i} \tag{42a}$$

$$= \frac{P(s_0 = i|\mathbf{O}, \theta)}{P(\mathbf{O}, \theta)}, \tag{42b}$$

and finally

$$aF_i^n = \frac{z_i}{\sum_i z_i} \tag{43a}$$

$$= \frac{P(s_T = i|\mathbf{O}, \theta)}{P(\mathbf{O}, \theta)}. \tag{43b}$$

In the Hidden Markov modelling problem, the observable but incomplete data is the observation sequence. To achieve a more accurate but still tractable likelihood function, both the observation sequence (the observable data) and the state sequence through a Markov chain (the unobservable data) need to be invoked. The incomplete data are $Y = \{O_t\}$ and the complete data are $X = \{O_t, s_t\}$. The parameter vector θ consists of the HMM parameters. The relationship between X and Y is through an unobserved vector $\mathbf{z} = (\ \mathbf{z}_1, \cdots, \mathbf{z}_T\ )$ where $\mathbf{z}_t$ is the indicator vector of length N (N-state HMM) whose components are all zero except for one equal to unity indicating the unobserved state associated with O_t [18, 15]. The T indicators $(\ \mathbf{z}_1, \cdots, \mathbf{z}_T\ )$ are not independently and identically distributed, but are specified to follow a first-order Markov chain. The E-step and M-step have been derived in the discussion above.

The EM algorithm will then yield an iterative maximization technique of the conditional expectation of the log likelihood function in the complete data space, and producing a sequence of estimates of log likelihoods in the incomplete data space that is at least non-decreasing, even though a global maximum may not be obtained. Specifically, the E-step calculates the log likelihood of the observation sequence, and the M-step yields the estimates of the state transition terms (aF, π, A) and output pdf B parameters separately.

3.3 EM formulation for VQ bigram HMM (VQBHMM)

The EM framework of conventional HMM reestimation in the previous subsection 3.2 is here extended for VQBHMM. (32) becomes (also refer to (25)):

$$\log P(s, \mathbf{O}|\theta^{\mathbf{n}}) = \log \pi_{s_0}^n + \sum_{t=1}^{T} \log a_{s_{t-1}s_t}^n + \sum_{t=1}^{T} \log b_{s_t}^n(O_t|O_{t-1}) + \log aF_{s_T}^n. \tag{44}$$

Using similar arguments to the ones in the subsection above, (33) is modified into:

$$Q_1 = \qquad \ldots \qquad + \tag{45a}$$
$$\sum_{j=1}^{N}\sum_{k=1}^{M}\sum_{m=1}^{M} D_{jkm} \log b_j^n(k|m)$$

where

$$D_{jkm} = \sum_{s} P(s, \mathbf{O}|\theta) M_{jkm}(s), \tag{46}$$

and $M_{jkm}(s)$ is the number of times symbol k is generated from state j conditional on symbol m occuring at the previous time instant. All the other state transition terms remain unchanged.

Then, the equivalent equation (41) for the conditional output process probability term becomes:

$$b_j^n(k|m) = \frac{D_{jkm}}{\sum_k \sum_m D_{jkm}} \tag{47a}$$

$$= \frac{P(O_t = k|s_t = j, O_{t-1} = m, \mathbf{O}, \theta)}{P(s_t = j|\mathbf{O}, \theta)}. \tag{47b}$$

Again, observe how the decoupling of the state transition terms and conditional output process probabilities allow each term to be treated separately.

4 VQ bigram HMM reestimation equations

In this section, the reestimation equations involved in the training phase of VQBHMMs are presented. The multiple codebook case is also considered. Finally, an alternative derivation of the reestimation equations is demonstrated using Lagrangian techniques.

4.1 Detailed implementation of VQBHMM reestimates

The empirical VQ bigram probabilities are estimated in the following way for each speech unit and used to initialise the state-dependent conditional output process probabilities by:

$$\text{bigram}_{i|j} = \frac{N_{i|j}}{\sum_j N_{i|j}}, \tag{48}$$

where $N_{i|j}$ denotes the number of occurences of VQ symbol i given that the previous VQ symbol is j.

For R observation sequences of various lengths, the modified Q-function (29) is now:

$$Q = \sum_{r=1}^{R} \sum_{s} P(s, \mathbf{O}|\theta) \log P(s, \mathbf{O}|\theta^{\mathbf{n}}). \tag{49}$$

The reestimation equations below are meant for one observation sequence and can easily be extended for multiple observation sequences since these are independent of each other. This implies that all the quantities in the reestimation equations are indexed by r, the particular utterance and summed across R, the total number of utterances present in the training of the models.

The following recurrence equations are invoked [23]. Define the forward partial probability as:

$$\alpha_t(i) = P(O_1, O_2, \ldots, O_t, s_t = i|O_{t-1}, \theta), \tag{50}$$

and the backward partial probability as:

$$\beta_t(i) = P(O_{t+1}, O_{t+2}, \ldots, O_T|s_t = i, O_t, \theta). \tag{51}$$

Scaling of those α and β terms are required to avoid underflow. The usual approach is through dividing all the probabilities by the sum of all the α terms at a particular time frame after that time frame has been processed in the forward pass. The same scaling values can be used in the backward pass. Alternatively, log arithmetic can be used to overcome underflow.

The forward recursions are initialized by:

$$\text{Initialization}: \begin{cases} \alpha_1^\star(i) = \pi_i b_i(O_1|O_0) & \forall i = 1, N \\ c_1 = \sum_{i=1}^{N} \alpha_1^\star(i) & \\ \alpha_1'(i) = \frac{\alpha_1^\star(i)}{c_1} & \forall i = 1, N \end{cases} \tag{52}$$

where O_0 is a special starting symbol.

The prime indicates scaled values and the starred quantities are intermediate calculated values.

$$\text{Forward recursion} \qquad \forall \mathrm{t} = 2, \mathrm{T}: \begin{cases} \alpha_t^\star(i) = \sum_{i=1}^N \alpha'_{t-1}(i) a_{ij} b_j(O_t|O_{t-1}) & \forall i = 1, N \\ c_t = \sum_{i=1}^N \alpha_t^\star(i) \\ \alpha'_t(i) = \frac{\alpha_t^\star(i)}{c_t} & \forall i = 1, N \end{cases} \tag{53}$$

The backward recursions are started off by:

$$\text{Initialization}: \begin{cases} \beta_T^\star(i) = aF(i) & \forall i = 1, N \\ \beta'_T(i) = \frac{\beta_T^\star(i)}{c_T} & \forall i = 1, N \end{cases} \tag{54}$$

$$\text{Backward Recursion} \qquad \forall \mathrm{t} = \mathrm{T}-1, 1: \begin{cases} \beta_t^\star(i) = \sum_{j=1}^N \beta'_{t+1}(j) a_{ij} b_j(O_{t+1}|O_t) & \forall i = 1, N \\ \beta'_t(i) = \frac{\beta_t^\star(i)}{c_t} & \forall i = 1, N \end{cases} \tag{55}$$

The E-step involves calculating the log likelihood of the observation sequence (27). It is easily carried out by using the scaled values [22] (page 1052) at each time frame:

$$\log P(\mathbf{O}|\theta) = -\sum_{t=1}^{T} \log c_t. \tag{56}$$

Now, define $\epsilon_t(i,j)$ as the number of transitions from state i to state j at time t, conditioned on the observation sequence, and is given by

$$\epsilon_t(i,j) = \alpha_t(i) a_{ij} b_j(O_{t+1}|O_t) \beta_{t+1}(j), \tag{57}$$

and $\epsilon_t(i)$ as the the number of transitions out of state i at time t, conditioned on the observation sequence, and is given by

$$\epsilon_t(i) = \alpha_t(i) \beta_t(i). \tag{58}$$

$\epsilon_t(i,j)$ and $\epsilon_t(i)$ can be replaced in terms of the scaled quantities:

$$\epsilon_t^\star(i,j) = \alpha'_t(i) a_{ij} b_j(O_{t+1}|O_t) \beta'_{t+1}(j), \tag{59}$$

and

$$\epsilon_t^\star(i) = \alpha'_t(i) \beta'_t(i). \tag{60}$$

It can easily be shown that the reestimation formulas remain unchanged.

From (47), the conditional output probabilities are given by:

$$b_i^n(k|m) = \frac{\sum_{t=1}^T \delta(O_t,k)\delta(O_{t-1},m)\epsilon_t(i)}{\sum_{t=1}^T \epsilon_t(i)} \tag{61a}$$

$$= \frac{\sum_{t=1}^T \delta(O_t,k)\delta(O_{t-1},m)\epsilon_t^\star(i)}{\sum_{t=1}^T \epsilon_t^\star(i)} \tag{61b}$$

$$= \frac{\sum_{t=1}^T \delta(O_t,k)\delta(O_{t-1},m)\alpha'_t(i)\beta'_t(i)}{\sum_{t=1}^T \alpha'_t(i)\beta'_t(i)} \qquad i = 1, N \qquad k, m = 1, M. \tag{61c}$$

From (40), the state transition reestimates are given by:

$$a_{ij}^n = \frac{\sum_{t=1}^{T-1} \epsilon_t(i,j)}{\sum_{t=1}^{T-1} \sum_{j=1}^{N} \epsilon_t(i,j)} \tag{62a}$$

$$= \frac{\sum_{t=1}^{T-1} \epsilon_t^\star(i,j)}{\sum_{t=1}^{T-1} \sum_{j=1}^{N} \epsilon_t^\star(i,j)} \tag{62b}$$

$$= \frac{\sum_{t=1}^{T-1} \alpha_t'(i) a_{ij} b_j(O_{t+1}|O_t) \beta_{t+1}'(j)}{\sum_{t=1}^{T} \alpha_t'(i)\beta_t'(i)} \qquad i = 1, N \qquad j = 1, N. \tag{62c}$$

The initial state transition vector is given, from (42), by:

$$\pi_i^n = \frac{\epsilon_1(i)}{\sum_{t=1}^{T} \epsilon_t(i)} \tag{63a}$$

$$= \frac{\epsilon_1^\star(i)}{\sum_{t=1}^{T} \epsilon_t^\star(i)} \tag{63b}$$

$$= \frac{\alpha_1'(i)\beta_1'(i)}{\sum_{t=1}^{T} \alpha_t'(i)\beta_t'(i)} \qquad i = 1, N, \tag{63c}$$

and from (43), the final state transition vector is:

$$aF_i^n = \frac{\epsilon_T(i)}{\sum_{t=1}^{T} \epsilon_t(i)} \tag{64a}$$

$$= \frac{\epsilon_T^\star(i)}{\sum_{t=1}^{T} \epsilon_t^\star(i)} \tag{64b}$$

$$= \frac{\alpha_T'(i)\beta_T'(i)}{\sum_{t=1}^{T} \alpha_t'(i)\beta_t'(i)} \qquad i = 1, N, \tag{64c}$$

The new output symbol probability has the intuitive interpretation that at state i, for the k-th symbol given a previous m-th symbol is observed, it is the expected number of times that the k-th symbol is output from the state given a previous m-th symbol is observed divided by the expected number of times in the state.

Training of such models from continuous speech material requires the following modifications. Each word is instantiated with its model which in turn is obtained from a concatenation of subword models. This large concatenated HMM is trained over the entire sentence. No time-aligned data (i.e. explicit time boundaries) is required. The embedded reestimation equations are essentially the same but more attention has to be paid to the state transition matrix of the concatenated HMM structure aligned to the whole sentence when mapped down again into the constituent phone HMMs.

4.2 Multiple codebook extension

A generalization of the observation process includes not only one VQ index at a given time t but multiple VQ indices from separately derived codebooks [24]. The differential codebook encodes information about the dynamics of cepstral changes. The power codebook is used mainly to distinguish between different types of voicing. The output observation term is calculated by multiplying the output probabilities of the individual sub-feature space. A much sharper distribution results from combination of three codebooks. Just like conventional HMMs can accomodate multiple codebooks, the same technique can easily be generalized to the VQBHMMs. By assuming that the multiple symbols at each time frame are independent from each other, (21) becomes:

$$P(\mathbf{O}|s,\theta) = \prod_{y=1}^{Y}\prod_{t=1}^{T} b_{s_t}^{y}(O_t^y|O_{t-1}^y) \tag{65}$$

where $\mathbf{O}$ is now the string sequence $\{O_1^1 \dots O_1^y \dots O_1^Y \dots O_t^y \dots\}$, Y is the total number of codebooks and $b_{jkm}^y \equiv P(O_t^y = k|s_t = j, O_{t-1}^y = m)$.

(32) is then modified to:

$$\log P(s,\mathbf{O}|\theta^{\mathbf{n}}) = \log \pi_{s_0}^n + \sum_{t=1}^{T}\log a_{s_{t-1}s_t}^n + \sum_{y=1}^{Y}\sum_{t=1}^{T}\log b_{s_t}^n(y)(O_t^y|O_{t-1}^y) + \log aF_{s_T}^n \tag{66}$$

which leads to the modified Q_1 being:

$$\begin{aligned} Q_1 \quad = \quad & \dots \quad + \\ & \sum_{y=1}^{Y}\sum_{j=1}^{N}\sum_{k=1}^{M}\sum_{m=1}^{M} D_{jkm}^y \log b_j^{n(y)}(k|m) \end{aligned} \tag{67a}$$

where

$$D_{jkm}^y = \sum_s P(s,\mathbf{O}|\theta) M_{jkm}^y(s) \tag{68}$$

and $M_{jkm}^y(s)$ is the number of times symbol k from the y-th codebook is generated from state j conditional on symbol m occuring at the previous time instant.

The reestimation formula for the conditional B matrix in the n-th iteration is then:

$$b_j^{n(y)}(k|m) = \frac{D_{jkm}^y}{\sum_k\sum_m D_{jkm}^y} \tag{69a}$$

$$= \frac{P(O_t^y = k|s_t = j, O_{t-1}^y = m, \mathbf{O}, \theta)}{P(s_t = j|\mathbf{O},\theta)}. \tag{69b}$$

The VQ bigram reestimation equations are easily modified to cope with multiple codebooks. Wherever there is an output probability b-element, it is replaced by a product of the individual b-elements. For instance, the forward recurrence relation becomes:

$$\alpha_{t+1}(i) = \Big\{\sum_{j=1}^{N}\alpha_t(j)a_{ji}\Big\}\prod_{y=1}^{Y} b_i(O_{t+1}^y|O_t^y) \qquad i = 1,N \qquad t = 1,T-1 \tag{70}$$

and the B matrix for each codebook is reestimated by

$$b_i^{n(y)}(k|m) = \frac{\sum_t \delta(O_t^y,k)\delta(O_{t-1}^y,m)\alpha_t(i)\beta_t(i)}{\sum_{t=1}^{T}\alpha_t(i)\beta_t(i)} \qquad i = 1,N \qquad k,m = 1,M \qquad y = 1,Y. \tag{71}$$

4.3 Lagrange interpretation of VQBHMM

The VQBHMM parameter estimates can also be posed as a constrained optimization of $\log P(\mathbf{O}|\theta)$, and another viewpoint of the reestimation formulas can be derived by imposing Lagrange multipliers on the log likelihood of $P(\mathbf{O}|\theta)$.

The following positivity constraints are present in VQBHMM.

$$\sum_i \pi_i = 1, \qquad \sum_j a_{ij} = 1 \qquad \forall i, \qquad \sum_i aF_i = 1 \tag{72}$$

$$\sum_m b_j(k|m) = \sum_m b_{jkm} = 1 \qquad \forall j, k \tag{73}$$

The Lagrangian function is

$$Q = \log P(\mathbf{O}|\theta) + \lambda_\pi \Big(\sum_i \pi_i - 1\Big) + \tag{74a}$$

$$\lambda_{aF}\Big(\sum_i aF_i - 1\Big) + \sum_{s_i} \lambda_{s_i} \Big(\sum_j a_{ij} - 1\Big) + \sum_{t_{jk}} \lambda_{t_{jk}} \Big(\sum_m b_{jkm} - 1\Big) \tag{74b}$$

where the various λ's are Lagrange multipliers for each element of the parameter set. Optimization with respect to the state transition elements have already been dealt in the literature, e.g. [22].

Differentiation of the Lagrangian function (74) with respect to the conditional b probabilities yields:

$$\frac{\delta Q}{\delta b_{jkm}} = \frac{\delta \log P(\mathbf{O}|\theta)}{\delta b_{jkm}} + \lambda_{t_{jk}}. \tag{75}$$

From the alpha and beta definition,

$$P(\mathbf{O}|\theta) = \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) a_{ij} b_j(O_{t+1}|O_t) \beta_{t+1}(j) \qquad \text{for any } t. \tag{76}$$

Hence,

$$P(\mathbf{O}|\theta) = \sum_{i=1}^{N} \alpha_T(i) \beta_T(i). \tag{77}$$

Thus,

$$\frac{\delta P(\mathbf{O}|\theta)}{\delta b_{jkm}} = \sum_{x=1}^{N} \frac{\delta \alpha_T(x)}{\delta b_{jkm}} \beta_T(x). \tag{78}$$

Now, by definition of alpha term (52),

$$\alpha_T(x) = \sum_{l=1}^{N} \alpha_{T-1}(l) a_{lx} b_x(O_T|O_{T-1}). \tag{79}$$

So,

$$\frac{\delta \alpha_T(m)}{\delta b_{jkm}} = \sum_{l=1}^{N} \frac{\delta \alpha_{T-1}(l)}{\delta b_{jkm}} a_{lx} b_x(O_T|O_{T-1}) + \sum_{l=1}^{N} \alpha_{T-1}(l) a_{lx} \frac{\delta b_x(O_T|O_{T-1})}{\delta b_{jkm}}. \tag{80}$$

Substituting (80) into (78) gives:

$$\frac{\delta P(\mathbf{O}|\theta)}{\delta b_{jkm}} = G + H \tag{81}$$

where

$$G = \sum_{x=1}^{N}\sum_{l=1}^{N} \frac{\delta \alpha_{T-1}(l)}{\delta b_{jkm}} a_{lx} b_x(O_T|O_{T-1})\beta_T(x). \tag{82}$$

But, from the definition of the beta term (55),

$$\beta_{T-1}(l) = \sum_{x=1}^{N} a_{lx} b_x(O_T|O_{T-1})\beta_T(x), \tag{83}$$

thus reducing

$$G = \sum_{l=1}^{N} \frac{\delta \alpha_{T-1}(l)}{\delta b_{jkm}} \beta_{T-1}(l). \tag{84}$$

The second term is

$$H = \sum_{x=1}^{N}\sum_{l=1}^{N} \alpha_{T-1}(l) a_{lx} \frac{\delta b_x(O_T|O_{T-1})}{\delta b_{jkm}} \beta_T(x) = \Big\{\sum_{l=1}^{N} \alpha_{T-1}(l) a_{lj}\Big\}\beta_T(j). \tag{85}$$

Again, from the alpha definition (52),

$$\alpha_T(j) = \Big\{\sum_{l=1}^{N} \alpha_{T-1}(l) a_{lj}\Big\} b_j(O_T|O_{T-1}) \tag{86}$$

so,

$$H = \frac{\alpha_T(j)\beta_T(j)}{b_j(O_T|O_{T-1})}. \tag{87}$$

By induction, from (84) when the differentiation is carried out until time index 1 is reached this will lead to:

$$\frac{\delta \log P(\mathbf{O}|\theta)}{\delta b_{jkm}} = \frac{1}{b_{jkm}} \sum_t \alpha_t(j)\beta_t(j) = \frac{1}{b_{jkm}} \sum_t P(O_t = k, \mathbf{O}|s_t = j, O_{t-1} = m, \theta), \tag{88}$$

the last term being easily derived from the definition of the alpha and beta terms.

From (75) and the above (88) and using conditional probabilities

$$\sum_t \frac{P(O_t = k, \mathbf{O}|s_t = j, O_{t-1} = m, \theta)}{P(\mathbf{O}|\theta)} + \lambda_{t_{jk}} b_{jkm} = 0. \tag{89}$$

Summing over index m,

$$\sum_m \sum_t \frac{P(O_t = k, \mathbf{O}|s_t = j, O_{t-1} = m, \theta)}{P(\mathbf{O}|\theta)} + \lambda_{t_{jk}} \sum_m b_{jkm} = 0. \tag{90}$$

From the above two equations (89) and (90), we arrive at same estimate (61) in section 4:

$$b^n_{jkm} = \frac{P(O_t = k|s_t = j, O_{t-1} = m, \mathbf{O}, \theta)}{P(s_t = j|\mathbf{O}, \theta)} \tag{91}$$

Given that the same estimates in section 3 are derived for the other parameters of the model, this proves that if the reestimation of parameters do not modify their values, a local extrema of the log likelihood function has been attained.

5 The Viterbi algorithm for speech recognition

This section introduces the optimality principle on which most speech recognition algorithms are based and shows how language constraints are incorporated. The actual Viterbi algorithm applied to multiple codebook VQBHMMs and the traceback of the recognized string sequence are described. For what follows in this section, the use of 'word' is generic and refers to any defined speech unit and includes, for example, a phoneme.

Dynamic Programming (DP) is a mathematical programming technique for solving certain sequential optimization problems. Because of the sequential nature of speech and language events, DP can be used to obtain a solution to these problems by decomposing them into a series of smaller problems and solving them sequentially. The main references for Dynamic Programming are by Forney, Viterbi and Bellman [25, 26, 27]. Bellman's principle of optimality [27] states that if an optimum path from point A to point C passes through some point B, then the sub-paths from A to B and from B to C must also be optimal paths. Thus, if the optimal state sequence from $s(0)$ to $s(T)$ passes through state $s(t)$ at time t, then it includes, as a portion of it, the optimal state sequence from $s(0)$ to $s(t)$ i.e. a non-optimal partial path can never be part of the global optimal path.

The recognition process can be described as computing the probability $P(W|A)$ that any word string W corresponds to the acoustic signal A, and finding this word string having the maximum probability. Using Bayes's rule,

$$P(W|A) = \frac{P(W)\,P(A|W)}{P(A)}.$$

The probability of the acoustic model $P(A)$ is dependent on the amount of training material present and the prior knowledge used for the acoustic models. The language model is determined independently of the acoustic models – it depends on the amount of data and assumptions necessary for the practical computation of the language model. Combining these two probabilistic models in the between HMM word rules will lead to an imbalance in the accumulated likelihood. So there needs to be a gross equalization term called a penalty term to give equal weight to these two contributions.

The goal of this DP algorithm also called the one-pass algorithm or Viterbi decoding/alignment/algorithm is to determine the sequence of HMMs that best matches the test speech, while obeying some constraints on possible speech unit transitions. With a language model, it is possible to eliminate many candidates from consideration because, in effect, higher probabilities are assigned to more probable word sequences than alternative ones. In particular, the bigram grammar indicates the probability that one particular word will occur given a previous one. In phoneme recognition, those bigram probabilities are computed from the training data.

Compared to the conventional Viterbi algorithm, the assumption made that all observations are statistically independent must be relaxed for the case of VQBHMMs. This section describes the Viterbi algorithm for decoding word sequences using such models. The pseudo-C code presented below has been programmed 'as is'. The output probability term in the modified Viterbi equation is effectively a measure of the likelihood of occurrence of a codeword given its predecessor. An unlikely VQ bigram pair decreases the log likelihood compared to an existing one. Because of the single link from state 3 to the final non-emitting state and the single link from the initial non-emitting state to state 1 (figure 2), there is no need for network 'compilation' to calculate interword transition probabilies. The Viterbi decoding equations for standard HMMs follow exactly the same pattern in the flow and implementation, with the b-term conditional probabilities simply being replaced by conventional b-term probabilities.

A difficulty occurs if one wishes to assign probabilities to known but unobserved words (or any arbitrary speech units) for the calculation of bigram probabilities for the language component. The heuristic solution normally applied is that the probability of an unobserved word is similar to the probability of that word with a count of one. In initializing the VQ bigram probabilities, no such considerations apply in the training because only the observed VQ pairs will be trained. However, there are VQ indices that do not appear in the training set for a given model but are in fact likely to occur in the test sentences – hence the use of a threshold, as in the conventional case, to avoid zeros in the estimated conditional probabilities in the recognition mode. This is necessary to overcome limitations of insufficient training data by assigning reasonable non-zero probabilities to all events. Two threshold values have been used to counter the three contigencies of one or two of the VQ symbols not present in the VQ pair. In the event of the present and the conditional symbol both being absent, the threshold term is held at 1×10^{-6}. In the event of either the present or the conditional symbol being absent, the threshold term is fixed at 1×10^{-3}.

The Viterbi algorithm presented below is similar in spirit to the one applied to Dynamic Time Warping as a pattern matching technique [28, 29]. All the concepts for successfully transferring it to the HMM framework are present such as the within-word transition, word transition and backtracking operations. All the required notations is given below.

t	t-th frame of the test sequence
j	j-th state of a given HMM (reference unit)
v	v-th HMM
V	total number of HMM units in vocabulary (presently 39 context-independent phones)
T	total number of input frames in test utterance
N_v	total number of states for the v-th model with state index starting at 0
$b_j^{v(y)}(O_t^y \mid O_{t-1}^y)$	emission probability of symbol O_t^y from the y-th codebook (log value) at time t from state j of the v-th HMM conditional on the previous y-th codebook time symbol
a_{ij}^v	state transition probability from state i to state j for the v-th HMM
aF_i^v	state transition probability to final non-emitting state i for the v-th HMM
$\phi[v][j]$	accumulated score likelihood of the best path to grid point (t, j, v) for a given time t
$B[j][v]$	backpointer chaining the leading point to point (t, j, v) with the ending frame of the preceding unit for a given time
$Bi[j][v]$	temporary array
$T[t]$	from-HMM array: records HMM v with maximum probability score at its ending state $N_v - 1$ at time t
$F[t]$	from-frame array: points to ending test frame of the preceding unit referenced in $T[t]$
R	number of decoded HMM units

The initialization of the likelihood array and the traceback buffers consists of the following steps.

/* *initialization* */
for $v = 1, \cdots, V$
    $\phi[v][0] = \sum_y b_1^{v(y)}(O_1^y|O_0^y) \qquad B[0][v] = 0$
    for $j = 1, \cdots, N_v - 1$
        $\phi[v][j] = -\text{HUGE} \qquad B[j][v] = 0$
    end of loop j
end of loop v

/* *Main Body* */
for $t = 2, \cdots, T$
    for $v = 1, \cdots, V$
        /* within-HMM transition rules */
        for $j = 0, \cdots, N_v - 1$
            $\phi[v][j] = \max_i \Big\{ \phi[v][i] + a^v[i][j] \Big\} + \sum_y b_j^{v(y)}(O_t^y|O_{t-1}^y)$
        end of loop j
        for $i = 0, \cdots, N_v - 1$
            /* $i^\star$ is the state with best state score in HMM v */
            $i^\star[v] = \arg\max_i \phi[v][i]$
            $Bi[i][v] = B[i^\star[v]][v]$
        end of loop i
    end of loop v

    /* find which HMM unit has best end state score */
    for $v = 1, \cdots, V$
        $v^\star = \arg\max_v \phi[v][N_v - 1]$
    end of loop v

/* Update $T[t]$ */
$T[t] = v^\star$

    /* Check whether the state transition from last state to first state of another HMM unit is more likely than a within HMM state transition */
    for $v = 1, \cdots, V$
        if $(\phi[v^\star][N_v - 1] + aF^v[i^\star[v]] + \text{bigram}(v|v^\star) + \text{penalty term} > \phi[v][0])$
            $\phi[v][0] = \phi[v^\star][N_v - 1] + aF^v[i^\star[v]-] + \text{bigram}(v|v^\star) + \text{penalty term}$
            $Bi[0][v] = t - 1$
        else
            /* the path stays in unit v for state 0 */
            $Bi[0][v] = B[0][v]$
    end of loop v

/* Update $F[t]$ */
$F[t] = Bi[N_{v^\star}][v^\star]$

$B[j][v] = Bi[j][v] \qquad \forall j, v$
end of loop t

$$\phi[v][j] = \max_i \Big\{ \phi[v][i] + a^v[i][j] \Big\} + \sum_y b_j^{v(y)}(O_t^y | O_{t-1}^y) \qquad j = 0, N_v - 1 \tag{92}$$

(92) is the local likelihood computation for all active states in the word models. In the course of the algorithm, backpointer array $Bi[j][v]$ is set to the input frame number corresponding to the end state of the model previous to the v-th model which is determined by the best path ot the grid point (t, j, v).

In order to find the sequence of words, a backwards trace through the vector $F[t]$ holding the frame position and vector $T[t]$ holding the identity of the previous model at each time frame is performed. First, the final model is identified by finding the entry point of the HMM which maximises the probability of the observation sequence at the final time frame; from this, the entry point and identity of the previous model are found, and so the traceback continues until all the word boundaries have been found. The backtracking of variables consists of the following operations:

/* *Backtracking* */
$r = T, R = 0$
while $(r \neq 0)$
  get unit $T[r]$
  find preceding unit from-frame array $r = F[r]$
  increment the number of decoded units R
end of while loop
Pull the phoneme sequence $\{U_1, \cdots, U_R\}$

6 Experimental results

This section describes the basic set-up of a conventional phone HMM recognizer and its baseline performance is presented. Next, using VQBHMMs, the phone recognition performance is reported together with details specific to the implementation. The way in which the VQ bigram pairs are arranged for efficient storage and recalling is of particular importance. A final experiment using crude variable frame rate compression is carried out in an attempt to put more emphasis on the more widely varying acoustic regions of the speech.

6.1 Baseline phone experiment

The DARPA TIMIT database has been used in the subsequent phone recognition experiments [30]. Many phoneme classification experiments have been reported on this database, the most relevant to this work being Lee's [1]. Because of its widespread use, it will prove much more effective and less controversial to evaluate the significance of the subsequent recognition results.

The December 1988 TIMIT acoustic/phonetic database consists of 420 speakers, 10 sentences per speaker, of which there are:

1. 2 'sa' calibration sentences spoken by all speakers
2. 5 'sx' phonetically balanced sentences derived from a list of 450 sentences
3. 3 'si' randomly selected sentences

The whole database has been segmented using 64 phoneme categories.

It is believed [1] that all the sentences used in this study for training and testing are amongst those used by Lee in his experiments – in fact about 300 training sentences less have been used. This is done to coincide with the training set used by Robinson [14]. TIMIT 'sa' calibration sentences are disregarded for either training or testing. Around 2500 sentences across all the dialects from 325 speakers are used as training data, and 160 sentences from 20 speakers across all the eight dialects are used as testing material.

The prevailing preprocessing conditions are:

- TIMIT sampling frequency 16 KHz
- Pre-emphasis coefficient 0.95
- frame length of 256 samples (16 ms)
- frame separation of 128 samples (8 ms)
- Hamming window

The preprocessing consists of the following components:

1. 10 dimension mfcc vector [31]
2. 10 dimension delta-mfcc vector [6], four frames separation
3. 1 dimension normalized power term [32], mainly to distinguish between voiced and unvoiced sounds

A vector quantization using Linde's algorithm [33] using about 350 000 vectors, independently for each preprocessor, to generate 256 prototypes is carried out. A 39 context-independent phoneme set is used and corresponds exactly to the one used by Lee [1]. There are 6047 symbols in the test set. The speakers involved in the test have the identities shown in table 1.

fdmy0	mtkd0	fjlr0	fkbw0	mslb0
fntb0	mrlr0	futb0	mtwh0	mbjv0
mdem0	fsem0	mdlm0	mdss0	mdsj0
mfwk0	mjee0	mpam0	mpfu0	mtjs0

Table 1: Identity of speakers in the test set of the TIMIT database.

[1] from private e-mail correspondence

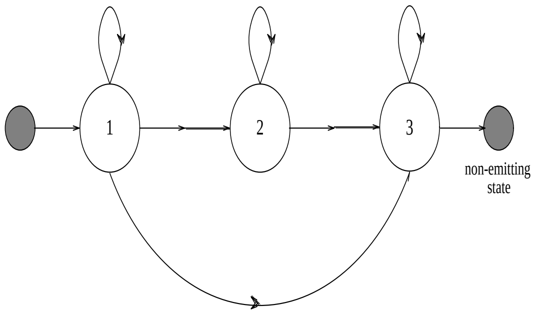

Figure 2: 3 state HMM used in implementation.

3-state HMMs are used by convention – better results may be obtained from using higher number of states with tied output probability density functions, but this avenue has not been investigated. The HMM structure used is illustrated in figure 2. By definition, for the model represented, there must be at least two frames associated with a token from that phoneme. But for certain phonemes, especially the unvoiced consonants like /p/, there is a significant amount of occurences of single-frame phonemes (about one-eight of the total training tokens for phoneme /p/ for the current data rate). This single VQ index is then replicated to conform to the physical requirement of the above model. The idea is borrowed from the stochastic segment model paper [3]. Each phoneme HMM is trained from its own time-aligned boundaries and 3 iterations are allowed. Embedded training is followed with 3 iterations over all the training sentences, reestimating after each iteration. A simple thresholding is applied to the resultant $'b'$ parameters for unobserved VQ symbols, with a floor value of $1\text{x}10^{-5}$ being used. A phone bigram language model is used in the Viterbi decoding algorithm with the probabilities determined from the training data. The DP-string matching process currently used has an insertion penalty term of 7, substitution being 10 and finally deletion 7 [34].

The phoneme recognition experiments are carried out from preprocessing to final recognition with the begin and end silences excised from the sentences, but the decoded strings and the actual phoneme strings are modified as if the two silence regions are recognized perfectly. Rigorously speaking, the penalty term should be adjusted on the whole training set to achieve a 10-12% insertion rate or any other reasonable criteria like equal insertion and deletion rates, Then, this particular penalty value must be used on the test set and whatever resulting proportion of insertion, deletion or substitution achieved recorded. A shortcut is done in that the penalty term is adjusted from the testing set itself to attain the desired insertion rate. The following baseline results have been achieved in table 2 and are about at the same level obtained by Lee [1].

preprocessor	correct (%)	insertion (%)	substitution (%)	deletion (%)	accuracy (%)
mfcc	50.5	10.0	35.4	14.1	40.6
delta-mfcc	45.4	11.5	38.1	16.6	33.9
power	32.3	12.1	48.5	19.2	20.1
3 codebooks	62.9	10.5	26.8	10.3	52.4

Table 2: Baseline percentage phone recognition results using standard phoneme HMMs on the TIMIT database.

6.2 VQ bigram phone recognition experiment

For mfcc, the co-occurence matrix entries will tend to have a smaller number of entries while for delta-mfcc the matrix will contain a larger number of entries because of the inherent variability associated with differential information. As for power, it is obvious from inspection that there is usually no consistent pattern of power VQ values across the different sentences. To fix ideas, for phoneme /aa/, in the whole training set, there are 3137 mfcc VQ pairs, 5349 delta-mfcc pairs and 8156 power VQ pairs. For phoneme /p/, in the same order is found 1476, 2435 and 3970. For phoneme /z/, it is 2156, 3848 and 5441 respectively.

An attempt can be made to replace the array of (m, n) grid points (VQ pairs) for a given state by a list. But then the list would have to be searched for each new VQ pair. This is inefficient as the worst case implies going through the whole list in addition to the overhead proper involved in the ordering and searching of the lists.

Instead, arrays that are explicitly addressed to a given VQ codeword will be used. An entry i in the list of VQ indices consists of the number of conditional VQ indices given codeword i has occured $n(i)$ times, and two pointers $b(i)$ and $e(i)$ that covers the range within which the conditional VQ indices and their corresponding probabilities are stored (See figure 3). This data-driven organization of the existing VQ pairs [2] in the training material greatly facilitates the storage and search for a given VQ pair in either recognition or training mode.

The way the data is organized is similar to the scheme used by Ney for the dynamic programming beam search for large vocabulary continuous speech recognition and done in such a way that there is no limit to the number of word hypotheses during the network search [35]. Essentially, the data structure consists of a list of words and a list of grid points. The list of words comprise of the word index and two pointers to the list of grid points bracketing the path hypotheses. The entry in the list of grid points consists of the state index, probability score and backpointer.

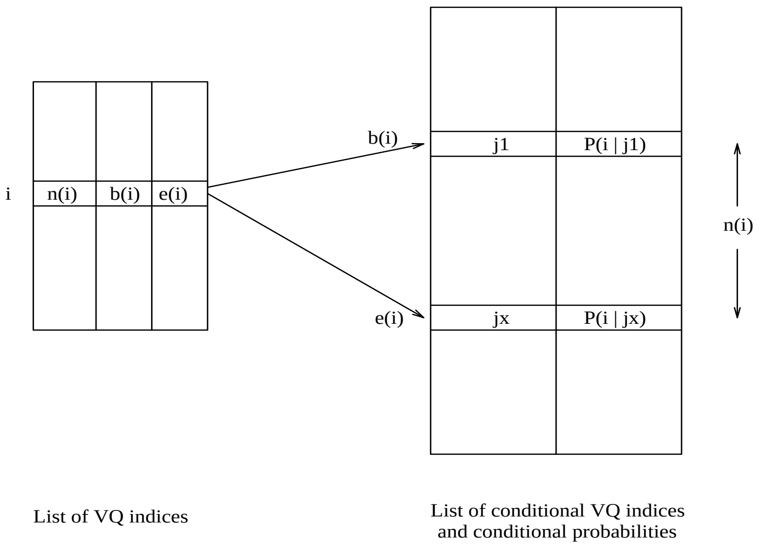

Figure 3: Implementation of storage and search for VQ pairs $P(i|j)$.

[2]The same principle can be applied to conventional HMM because on average about half of the VQ bins are empty and therefore the reduction of memory for B matrices is about half. But the most important consequence is the huge reduction that may be possible in any subsequent development into context dependent VQBHMMs.

In empirical terms, it is found that the amount of memory necessary to cope with VQBHMM is in the order of 15 times the amount used for standard HMMs, the computation required for training is about 7 times longer and finally the Viterbi decoding for VQBHMMs lasts about 5 times longer. Those values are highly approximate but provide a rough indication.

Each phoneme VQBHMM is initially trained by VQ sequences derived from its own time-aligned boundaries with a single closest VQ index taken from the adjoining left and right neighbour phonemes with the exception of the phoneme being the first or last one in the sentence. Initialization of the VQ bigram probabilities is very important as convergence of the log likelihood is much faster than with a uniform or random initialization. Typically, for practical convergence, 3 iterations are needed with proper initialization whereas the uniform or random seeding of probabilities requires about 10 iterations. This situation should be contrasted with the conventional HMM whereby random or uniform or *a priori* probabilities determined from the VQ data used to seed the B matrix element probabilities do not affect the practical convergence, usually achieved after about 3 iterations. Embedded training is the following stage, with 3 iterations carried out.

Quite a substantial proportion of existing VQ pairs (on average about 8 to 10%) are found to occur only a few times. After normalization, the probability is rather small of the order of 0.001. To reduce computation and ensure a more robust estimation of the parameters for a given amount of training material, VQ bigram pairs for a particular state occuring at probabilities less than 0.005 are eliminated from consideration in the training data. (A small experiment carried out indicates that no degradation in performance is noticed.) The following phone recognition performance (table 3) using VQBHMMs is obtained.

preprocessor	correct	insertion	substitution	deletion	accuracy
mfcc	56.1	10.7	31.6	12.4	45.4
delta-mfcc	49.4	11.3	34.7	15.9	38.1
power	33.2	10.9	48.1	18.7	22.2
3 codebooks	68.6	9.1	21.3	10.1	59.5

Table 3: VQBHMM phone recognition results on the TIMIT database.

It is encouraging to note that delta-mfcc VQBHMM gives some improvement and, to a lesser extent, VQ bigram power HMM, indicating indirectly that second-order differential information helps phoneme recognition. The mfcc results are close to that of the combined mfcc/delta-mfcc of Lee [1] (contrast 56.1% correct/45.4% accurate to Lee's 57.9%/47.9-45.9% accurate for two independent codebooks for cepstral and differential cepstral preprocessors) and somehow indicates that differential information is carried in the estimated mfcc VQ bigram probabilities. Individual improvements achieved by each codebook do not just add up when the 3 codebook case is considered because of the overlap in the improvements for each individual preprocessor VQ bigram model. Considering the percentage correct criterion, a 14% error rate reduction has been achieved.

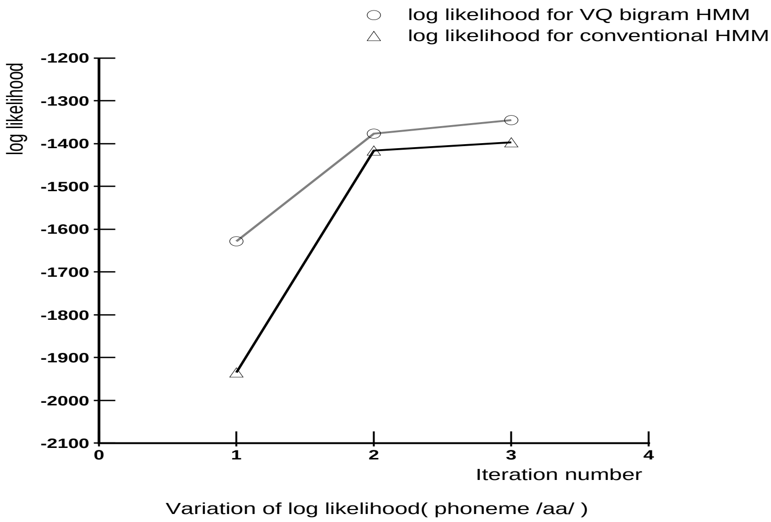

Figure 4: Variation of log likelihood with iteration number for conventional HMM and VQBHMM in Baum-Welch reestimation phase.

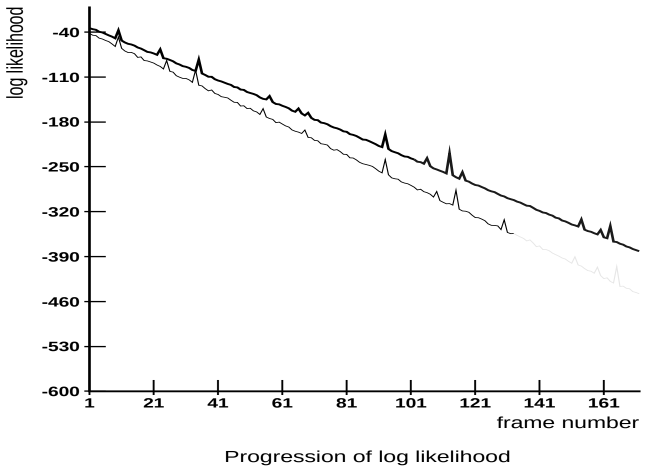

Figure 5: Progression of accumulated log likelihood score for conventional HMM and VQBHMM (the darker line is for VQBHMM and the lighter one is for conventional HMM) in Viterbi decoding testing phase.

An increase in absolute likelihood from the training data (refer to figure 4 for progression of the log likelihood in Baum-Welch training for phoneme /aa/) for the conditional model relative to the standard HMM does not necessarily mean an increase in recognition performance on the test set. However, it does imply a more reliable prediction and more accurate acoustic modelling of the given training data. The Viterbi score progression for sentence dr2/mdem0/sx158 (refer to figure 5) shows as further empirical evidence the higher acoustic modelling capabilities of VQBHMM than the conventional HMM. The generic assumption here is that if a given model provides a better fit in an ML sense to a training set of words than alternative models, it should then provide better fits to an independent test set of words, and hence improve recognition performance.

From Bayes' rule, the posterior probabilities for a given word sequence $\mathbf{W}$ are given by

$$P(\mathbf{W}|\mathbf{O},\theta) = \frac{P(\mathbf{O}|\mathbf{W},\theta)P(\mathbf{W}|\theta)}{P(\mathbf{O}|\theta)}. \tag{93}$$

In the training phase, $P(\mathbf{O}|\mathbf{W},\theta)$ is maximized whereas in recognition the posterior probabilities $P(\mathbf{W}|\mathbf{O},\theta)$ are required. $P(\mathbf{O}|\theta)$ is a constant during recognition and $P(\mathbf{W}|\theta)$ is the bigram phone language model. In general, for a conventional HMM, the equation below is at the heart of the Viterbi formulation for inter-model transitions (Remember that the penalty term governs the rate of recognized phonemes to the actual phoneme length. The insertion rate is restricted to lie between 10-12%):

$$P(\mathbf{W}|\mathbf{O},\theta) = P(\mathbf{O}|\mathbf{W},\theta)P(\mathbf{W}|\theta)P(\text{HMM weighting factor}) \tag{94}$$

For a VQBHMM model, (94) becomes

$$P(\mathbf{W}|\mathbf{O},\theta) = P(\mathbf{O}|\mathbf{W},\theta)P(\mathbf{W}|\theta)P(\text{VQBHMM weighting factor}) \tag{95}$$

Now, it has been shown earlier that $P(\mathbf{O}|\mathbf{W},\theta)$ is larger in magnitude for VQBHMM than the corresponding conventional HMM (i.e. the likelihood of the acoustic evidence given the word sequence is higher for VQBHMM than for conventional HMMs) and because the phone bigram language model is the same for both cases, the weighting term for a specified insertion rate has to be smaller for the VQBHMM model than its counterpart. This has been verified in the experiments where this weighting term is found to be about -18.0 for conventional HMM and -32.0 for VQBHMM models for the three codebook case.

6.3 A simple variable frame rate experiment

The possible negative influence of long periods of fairly stationary speech to the accumulated likelihood in the Viterbi algorithm can be minimized by reducing the frame rate, but at the same time retaining parts of the signal which are changing rapidly. There are several ways of selectively emphasizing the speech signal and can be collectively called variable frame rate (vfr) analysis [3, 36]. After using vfr, the fast transitions are weighted more heavily than steady-state regions.

However, vfr can be carried in a crude way by considering strings of 3 successive VQ indices which are identical in the mfcc observation sequence and compressing them into one VQ index. At the corresponding time instant of compression, the VQ indices in the differential mfcc and power observation sequence are missed out. This exceedingly simple compression has been done to the mfcc preprocessor because of all the three preprocessors, it is the one with the least variability in signal structure. On average, about 94% of the original string length remains after this crude frame rate compression.

preprocessor	correct	insertion	substitution	deletion	accuracy
3 codebooks	61.1	10.3	26.5	12.4	50.8

Table 4: Phone recognition results for a simple vfr analysis using VQBHMM.

This crude vfr analysis does not yield any improvements (table 4). In fact, the recognition results deriorates slightly. It could be that patterns of interdependencies do not change substantially with the duration of the sentence. The recognition results, likewise, show little change. For VQBHMMs, like conventional HMMs, the averaging property of the reestimation equations seems to swamp out the effect of the transient regions of the speech signal because of their short durations.

7 Discussion

This section contains general conclusions regarding the use of VQBHMMs in modelling time sequences of measurement vectors. Possible extensions of the method for increased phone accuracy are mentioned.

An attempt has been made to reduce the effect of modelling inaccuracies present in conventional HMM paradigm, more specifically in the output-independence assumption. More accurate temporal information has been incorporated in the output probability process by considering the relative likelihoods that certain VQ codewords follow others, given a certain unobservable sequence in the state progression of the HMM state network. This implies that an unlikely occurence of VQ bigram indices decreases the accumulated log likelihood in the Viterbi scoring. Reestimation equations of those VQ bigram probabilities from each HMM state have resulted from a simple variant of the EM algorithm as applied to the standard HMMs. Indeed, the EM framework is a very powerful technique which allows the various components of the complete data log likelihood to be decoupled into essentially the state transition and output process components. The E-step calculates the log likelihood of the observation sequence and the M-step yields the estimates of the state transition terms and conditional output pdf parameters separately.

A phoneme error rate reduction of about 14% has been achieved for the TIMIT continuous phoneme recognition task. Although VQBHMM has not given a big error rate reduction compared to for instance, triphone modelling using about 1000 generalized context-dependent models where an error rate reduction of 28% has been achieved [1], fewer parameters have been used. About 16% error rate reduction has been achieved using a tied-mixture HMM [37]. The database is speaker-dependent with 47 existing phonemes and the number of test symbols is 1748. 75% correct/69% accurate is obtained by using context information through a mechanism which adds some left context and right context information to the inputs of recurrent neural nets [14].

Normally the more parameters modelling the acoustic events, the greater the potential for higher recognition accuracy. However, for a given amount of training data, the variance associated with the estimate of those probabilities will be increased. This method of VQ bigram estimation may be less effective in limited training data conditions because these VQ bigram models are data-driven. A small sub-experiment has been carried out to demonstrate that the developed algorithms are still robust for fairly limited training data, consisting of single-speaker digit recognition with nine instances of each digit from a clean-speech car database for training and testing – conventional Hidden Markov modelling yields 100% accuracy as does the VQ bigram Hidden Markov modelling.

Several points associated in the implementation of VQBHMM for continuous phoneme recognition should be emphasized. A data-driven organization of the VQ bigram indices for each state and phoneme concerned has greatly facilitated access to the corresponding probabilities for a given VQ pair. Proper initialization of the conditional output probabilities considerably reduces the training time. Finally, when using the Viterbi decoding algorithm, another threshold term is required in the event that an incoming VQ pair from the test sentence has never been seen and estimated from the training set.

The likelihood of the observation sequence has been shown to be higher for VQ bigram Hidden Markov modelling than for conventional Hidden Markov modelling. However, by itself, this does not imply more refined discrimination among the different classes, but more accurate in-class acoustic modelling. The improvement obtained from the differential mfcc supports the use of second-order temporal differences which have recently been in the literature, e.g. Ney's paper [38], and improving recognition results. It must be emphasized throughout that the training and recognition algorithms are not excessively more complex than the conventional HMM ones, and simply requires the replacement of the emission probabilites with the correlated ones.

7.1 Future work

Just like trigrams are sometimes used in grammar modelling, the VQ bigram structure can be expanded into a VQ trigram HMM where the output independence assumption is relaxed further and now becomes the probability of emission of a particular symbol conditional on other symbols occuring at the two previous time instants from a given state. A possible tied-mixture (semi-continuous) [37, 39] VQBHMM can be formulated. Each codeword of the VQ codebook will be represented by a continuous pdf.

VQBHMM can be used in conjunction with all other improved duration modelling techniques without any modification to the estimates of the conditional output probability process. The duration modelling capability of the A matrix is unsophisticated – in the current model this is governed by 7 parameters for each phoneme. Duration-defined models, whereby the training data for a given phoneme is divided approximately into 3 groups depending on their durations, have given improved phone recognition accuracy almost to the same level [3] (67.5% correct is achieved with 9.8% insertion rate) that has been achieved by the VQBHMM. The B matrix for each phoneme is tied for each of those three models. Better still, the additional memory requirements are negligible – 39 x 9 x 2 additional parameters. The data has simply been divided into three equal groups in increasing order of duration ranges. Extension of this technique to the VQBHMM should be straightforward. As the number of models per phoneme is increased, the simplified duration clustering procedure does a better job at modelling the within-class data. A similar idea has been attempted by Deng but only vowels were considered [40].

Unfortunately, the gain in improvement achieved by context independent VQBHMM has not been to the level reached by generalized triphone HMMs for recognition to be carried out at a word level. A particular phoneme has been assumed throughout to be unaffected by other neighbouring phonemes, a very crude assumption indeed. Triphone modelling is very

[3] All the previous experiments were carried out with the silences at the beginning and end of the sentences excised. When the experiments are repeated with silences restricted to a maximum of 160 ms (about 18 frames) at both ends, the recognition results are marginally better than the ones achieved by Lee with 65.1% correct with 10.4% insertion rate. Note that the training set now consists of about 2800 sentences. The most probable reason for the discrepancy between the results shown in table 2 and the aforementioned results can be ascribed to a more accurate coverage in the codebook of the silence regions of speech and beginnings of burst-like fricatives and closures. The codebook training size data is increased by roughly 56,000 vectors – about 15% of the original training size data.

powerful if properly carried out since it models the most immediate co-articulatory effect. In Lee's work [41], it is found that context-dependent modelling reduces the word error rate by as much as 60%. Similarly, Schwartz found out that for both phone and word recognition, modelling phone-in-context reduces the error rate significantly by about 50% [42]. Along the lines of Schwartz's work the algorithm can be extended for context dependent VQ bigram Hidden Markov modelling.

8 Acknowledgements

One of the author is indebted to Trinity Hall for a scholarship. I wish to thank members of the Speech group for advice, especially Patrick Gosling for logistic help and Tony Robinson for immediate response on the TIMIT database and related questions in speech recognition as well as the use of his DP string alignment algorithm.

References

[1] K. F. Lee and H. W. Hon. Speaker-independent phone recognition using hidden Markov models. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(11):1641–1648, November 1989.

[2] R. Schwartz et al. Robust smoothing methods for discrete hidden Markov models. In *Proc. ICASSP*, pages 548–551, 1989.

[3] M. Ostendorf and S. Roucos. A stochastic segment model for phoneme-based continuous speech recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(12):1857–1869, December 1989.

[4] C. H. Lee. On the use of some robust modeling techniques for speech recognition. *Computer Speech and Language*, 3:35–52, 1989.

[5] A. Kriouile, J. F. Hari, and J. P. Haton. Some improvements in speech recognition algorithms based on hidden Markov model. In *Proc. ICASSP*, pages 545–549, 1990.

[6] S. Furui. Speaker-independent isolated word recognition using dynamic features of the spectrum. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 34(1):52–59, February 1986.

[7] S. Huang and R. M. Gray. Conditional histogram vector quantization for spellmode recognizer. In *Proc. ICASSP*, pages 1930–1933, 1987.

[8] E. Loeb and R. F. Lyon. Experiments in isolated digit recognition with a cochlear model. In *Proc. ICASSP*, pages 1131–1135, 1987.

[9] K. Sugarawa et al. Isolated word recognition using hidden Markov models. In *Proc. ICASSP*, pages 1–4, 1985.

[10] H. Hattori. Speaker adaptation based on Markov modeling of speakers in speaker-independent speech recognition. In *Proc. ICASSP*, pages 845–849, 1991.

[11] A. Nadas, D. Nahamoo, and M. A. Picheny. Speech recognition using noise-adaptive prototypes. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(10):1495–1503, October 1989.

[12] P. Brown. *The Acoustic-Modeling Problem in Automatic Speech Recognition.* PhD thesis, Carnegie Mellon University, 1987.

[13] C. J. Wellekens. Explicit time correlation in hidden Markov models for speech recognition. In *Proc. ICASSP*, pages 384–386, 1987.

[14] T. Robinson and F. Fallside. Phoneme recognition from the TIMIT database using recurrent error propagation networks. Technical Report CUED/F-INFENG/TR.42, Cambridge University Engineering Department, 1990.

[15] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal Royal Statistics Soc. Ser. B (methodological)*, 39:1–38, 1977.

[16] R. A. Redner and H. F. Walker. Mixture densities, maximum likelihood and the EM algorithm. *Society For Industrial and Applied Mathematics Review*, 26(2):195–239, April 1984.

[17] L. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occuring in the statistical analysis of probabilistic functions of Markov chains. *Annals of Mathematical Statistics*, 41:164–171, 1970.

[18] L. E. Baum. An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes. *Inequalities*, 3:1–8, 1972.

[19] L. A. Liporace. Maximum likelihood estimation for multivariate observations of Markov sources. *IEEE Transactions on Information Theory*, IT-28:729–734, September 1982.

[20] E. L. Lehmann. *Theory of Point Estimation.* Wadsworth & Brookes/Cole. Statistics/Probability series, 1991.

[21] C. F. J. Wu. On the convergence properties of the EM algorithm. *The Annals of Statistics*, 11:95–103, 1983.

[22] S. E. Levinson, L. R. Rabiner, and M. M. Sondhi. An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 62(4):1035–1074, April 1983.

[23] Lalit R. Bahl, Frederick Jelinek, and Robert L. Mercer. A maximum likelihood approach to continuous speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(2):179–190, March 1983.

[24] V. N. Gupta, M. Lennig, and P. Mermelstein. Integration of acoustic information in a large vocabulary word recognizer. In *Proc. ICASSP*, pages 697–700, 1987.

[25] G. D. Forney. The Viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, March 1973.

[26] A. J. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13:260–269, April 1967.

[27] R. Bellman and S. Dreyfus. *Applied Dynamic Programming*. Princeton University Press, 1962.

[28] H. Ney. The use of a one-stage dynamic programming algorithm for connected word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(2):263–271, April 1984.

[29] C. Godin and P. Lockwood. DTW schemes for continuous speech recognition: a unified view. *Computer Speech and Language*, 3:169–198, 1989.

[30] John S. Garofolo. *Getting Started with the DARPA TIMIT CD-ROM: An Acoustic Phonetic Continuous Speech Database*. National Institute of Standards and Technology (NIST), Gaithersburgh, MD, 1988.

[31] S. B. Davis and P. Mermelstein. Comparison of parametric representation for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4):357–366, August 1980.

[32] V. N. Gupta et al. Using phoneme duration and energy contour information to improve large vocabulary isolated word recognition. In *Proc. ICASSP*, pages 341–345, 1991.

[33] Y. Linde, A. Buzo, and R. M. Gray. An algorithm for vector quantiser design. *IEEE Transactions on Communications*, 28:84–95, 1980.

[34] J. Picone et al. Automatic text alignment for speech system evaluation. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 34(4):780–784, August 1986.

[35] H. Ney, D. Mergel, A. Noll, and A. Paeseler. A data-driven organization of the dynamic programming beam search for continuous speech recognition. In *Proc. ICASSP*, pages 833–836, 1987.

[36] K. M. Ponting and S. M. Peeling. The use of variable frame rate analysis in speech recognition. *Computer Speech and Language*, 5:169–179, 1991.

[37] X. D. Huang and M. A. Jack. Semi-continuous hidden Markov models for speech signals. *Computer Speech and Language*, 3:239–251, 1986.

[38] H. Ney. Experiments on mixture-density phoneme modelling for the speaker-independent 1000 word speech recognition DARPA task. In *Proc. ICASSP*, pages 713–717, 1990.

[39] J. R. Bellegarda and D. Nahamoo. Tied mixture continuous parameter modelling for continuous speech recognition. In *Proc. ICASSP*, pages 2033–2045, 1990.

[40] L. Deng, M. Lennig, V. N. Gupta, and P. Mermelstein. Modeling acoustic-phonetic detail in an HMM-based large vocabulary speech recognizer. In *Proc. ICASSP*, pages 509–513, 1988.

[41] K. F. Lee. Context-dependent phonetic hidden Markov models for speaker-independent continuous speech recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 38(4):599–609, April 90.

[42] R. Schwartz et al. Context dependent modeling for acoustic-phonetic recognition of continuous speech. In *Proc. ICASSP*, pages 1205–1209, 1985.