A COMPARISON OF EXACT AND APPROXIMATE ALGORITHMS FOR DECODING AND TRAINING LOOSELY COUPLED HMMS

H.J. Nock Cambridge University Engineering Dept, Trumpington St, Cambridge CB2 1PZ.
(Currently visiting SSLI Lab, University of Washington, Seattle, USA.)

S.J. Young Cambridge University Engineering Dept, Trumpington St, Cambridge CB2 1PZ.

1 INTRODUCTION

Statistical speech recognition systems based on Hidden Markov Models (HMMs) have been successfully applied to the transcription of carefully dictated speech (eg. [33]). However, their performance degrades severely when confronted with conversational speech. For example, the DARPA HUB4E Broadcast News Evaluation [10] includes both spontaneous and more formal utterances in studio recording conditions: every system entered into the 1998 evaluation produced less accurate transcriptions for the spontaneous speech condition (F1) than for the more formal, planned studio speech condition (F0). [29] and [32] use a data set created by recording the *same* word-level transcript in a variety of speaking styles to confirm that conversational speech poses greater difficulty to current recognizers. It has been widely hypothesized that modelling schemes more sophisticated than HMMs will be required to achieve acceptable transcription performance on this type of data. Our current research is investigating a more general class of models, based upon loosely coupled or *factorial* HMMs [8][1]. These models are theoretically appropriate for modelling parallel loosely coupled time series, which for speech recognition tasks might comprise (for example) cepstra derived from frequency subbands or perhaps phonological or articulatory features. Since speech is produced by a system of loosely coupled articulators, stochastic models which explicitly represent this parallelism may have advantages for automatic speech recognition (ASR), particularly when trying to model the phonological effects inherent in casual spontaneous speech. Papers [22, 23] use a simple speech task to compare models in this family with more conventional speech models when using exact estimation and decoding algorithms. However, exact algorithms for general factorial models are potentially computationally expensive as the number of parallel time series increases or as the units over which asynchrony between time series is to be allowed become longer. Therefore, approximate schemes may be required to use these models for large vocabulary tasks.

The structure of the paper is as follows. Section 2 outlines the theory of factorial HMMs (FHMMs) and then describes the specific factorial model investigated in this work: the *Mixed-Memory Approximation to a FHMM* (MMFHMM). Nock [23] derives an EM algorithm and generalized forward-backward algorithm for exact MMFHMM estimation and likelihood calculations; Section 3 of this paper notes that efficiency of the forward-backward calculations can be improved for one potentially useful special case of the MMFHMM. Sections 4 and 5 describe approximate algorithms suitable for use in

[1] Papers investigating related models for speech applications include [17, 18, 35, 31, 16].

MMFHMM decoding alone or in both training and decoding. Section 6 compares the exact and approximate algorithms using a standard isolated word classification task. The paper ends with conclusions and outlines future research.

2 FACTORIAL HMMS AND THE MIXED-MEMORY APPROXIMATION

Suppose we wish to model K loosely coupled time series, where the observations in each time series (or *stream*) k, denoted $o_1^k, o_2^k, \ldots, o_T^k$, are produced on the same time-scale and may be scalars or vectors. These representations are assumed to comprise K loosely coupled time series, where each time series might correspond to a position trace for a particular articulator or to a phonological feature such as voicing, and observations in each time series are assumed to be produced on the same time scale.

One obvious scheme for modelling such data combines the K observations at each time t into a single observation vector $O_t = (o_t^1, \ldots, o_t^K)$ and then builds a standard HMM. However, the resulting model would not be a parsimonious representation of the data. Alternatively, we might model each stream k independently by using a single HMM per stream. The individual likelihood scores from the independent HMMs can be then combined in some fashion to obtain an overall score, as in the multiband framework (eg. [20, 14]). However, this scheme can capture only weak coupling between the different time series. An intermediate approach is to combine the K independent HMMs into a joint model which can capture more of the coupling between different streams. We can form a combined or *factorial* HMM in which (i) the hidden state space is the Cartesian product of the state spaces of the individual HMMs (see Figure 1), and (ii) the observations O_t are formed by concatenating the individual stream observations at time t, ie. $O_t = (o_t^1, \ldots, o_t^K)$. We refer to the Cartesian product hidden state space as the *metastate space*, to distinguish it from the state spaces of the original independent HMMs for each stream.

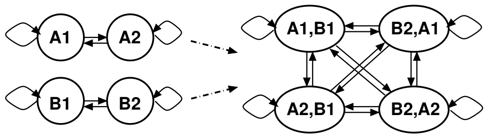

Figure 1: Metastate Space From Combined Ergodic HMMs A, B

The combined model as just described is equivalent to a standard HMM, in which the metastates and observations have internal structure. However, as K and N increase, estimation of a transition matrix and output densities for this combined or *factorial* HMM will be intractable both computationally and in terms of robust parameter estimation. Recent work in the machine learning literature handles these difficulties through additional conditional independence assumptions and approximations which exploit the internal, combinatorial structure of the metastates and observations both to reduce the number of parameters and as the basis for efficient, approximate training and decoding algorithms (eg. [8, 30]). As discussed in the next section, under appropriate parameter reduction schemes this general factorial or loosely coupled model contains several standard speech models as special cases, but also contains many more general models such as the MMFHMM[2].

2.1 Parameter Reduction Schemes

We assume that each time series k comprises D-dimensional observations o_t^k and combined observations are denoted $O_t = (o_t^1, \ldots, o_t^K)$[3]. We generalize the presentation in the previous section to allow L underlying Markov chains, where it is not necessary that $L = K$; hidden metastates therefore comprise L hidden variables and are described by L-tuples $I = (i^1, \ldots, i^L)$ and $J = (j^1, \ldots, j^L)$. Each chain has N possible states[4]. P denotes probability mass functions (pmf), p denotes densities. Using this notation, the parameters to be estimated in the factored model are the metastate priors $\pi(J)$, the metastate transition matrix $P(J|I)$ and the metastate observation distributions $p(O_t|J)$; the full set of MMFHMM parameters is denoted by λ.

Several conventional speech models are obtained under the following basic conditional independence (cond. indep.) assumptions:

- cond. indep. of initial metastate components: $\pi(J) = \prod_{k=1}^K \pi^k(j^k)$
- cond. indep. of metastate components given previous metastate: $P(J|I) = \prod_{k=1}^K P(j^k|I)$
- cond. indep. of observation components given current metastate: $p(O_t|J) = \prod_{k=1}^K p(o_t^k|J)$

By setting $K = L = 1$ in this parameter-reduced model, we obtain the standard *HMM*. If we set $L = 1$ and set K to the number of output streams, we obtain the *HTK synchronous multiple stream model* [34]. If we set $L = K$ and make the additional conditional independence assumptions that $p(j^k|I) = p(j^k|i^k)$ and $p(o_t^k|J) = p(o_t^k|j^k)$, then we obtain an asynchronous, *independent streams model* in which each time series is modelled by a separate HMM. This model is related to the multiband model of eg. [20]. However, our real interest in factorial models lies in investigating new parameter reduction schemes to obtain tractable models that are still powerful enough to effectively represent coupling between the K time series. Parameter reduction through the use of standard maximum likelihood parameter tying schemes (similar to eg. [26, 25]) is one possibility that we are investigating, but the work in this paper adopts the *Mixed-Memory Approximation* proposed in [30][5]:

- approximate the prior-related conditional probabilities with a convex combination of *cross-prior* matrices:

$$\pi^k(j^k) = \sum_{l=1}^K \psi^k(l)\pi^{kl}(j^k) \tag{1}$$

[2] Factorial models may also be motivated as instances of the more general family of Directed Graphical Models [15, 11].
[3] It is straightforward to allow observations of different dimensionality in each stream k.
[4] It is straightforward to allow different numbers of states per chain l.
[5] A similar approximation is used in [27].

- approximate the transition-related conditional probabilities with a convex combination of *cross-transition* matrices:

$$P(j^k|I) = \sum_{l=1}^{K} \psi^k(l) a^{kl}(j^k|i^l) \tag{2}$$

- approximate the observation-related conditional probabilities with a convex combination of *cross-emission* distributions:

$$p(o_t^k|J) = \sum_{l=1}^{K} \phi^k(l) b^{kl}(o_t^k|j^l) \tag{3}$$

The parameters $\pi^{kl}(j^k)$ are K^2 elementary $1 \times N$ cross-stream prior matrices, a total of K^2N prior parameters. The parameters $a^{kl}(j^k|i^l)$ are K^2 elementary $N \times N$ cross-transition matrices, a total of K^2N^2 transition parameters. The $b^{kl}(o_t^k|j^l)$ are K^2N cross-emission output densities; for D-dimensional observations and full covariance Gaussians, a total of $K^2ND(1+D)$ observation-related parameters. Parameters $\psi^k(l)$, $\phi^k(l)$ are mixture weights. They are fixed for a single model, and give a measure of the dependency between different streams, using a total of $2K^2$ parameters. This model thus has $\mathcal{O}(K^2(N^2 + ND^2))$ free parameters vs. $\mathcal{O}(N^K(N^K + K^2D^2))$ for the combined, full metastate space model.

Use of the Mixed-Memory Approximation allows separate evaluation of the effects of making transition-related or observation-related probabilities dependent upon full metastate identity, as well as the case where both are dependent upon the full metastate identity [22, 23]. Later sections of this paper refer to the special case of the MMFHMM in which only observation distributions are made dependent upon metastates and underlying chains are assumed independent: this *observation-only* MMFHMM (OMMFHMM) sets ψ to the $K \times K$ identity matrix.

2.2 Maximum Likelihood Estimation of the MMFHMM

Maximum Likelihood (ML) estimation of the MMFHMM may be achieved using the EM algorithm [7] (assuming appropriate choice of observation distribution). Following [30], equations (1)-(3) are viewed as mixture models, introducing two new types of latent variables x_t^k, y_t^k in addition to variables s_t^k denoting the metastate sequence taken through the model. The new latent variables encode the identity of the cross-emission distribution and cross-transition matrix (ie. the distributions within each mixture model) used in each stream k at each t. The information provided by the x_t^k and y_t^k variables is illustrated in Figure 2. The posterior probabilities necessary for the resulting parameter updates can be calculated using a (notationally complicated) generalization of the Forward-Backward Algorithm. A derivation of both, for the case of multivariate Gaussian observation distributions, may be found in [23].

Unfortunately, whilst the mixed-memory approximation does reduce the number of model parameters, exact likelihood calculations and EM-based estimation for the MMFHMM still require calculation of forward and backward probabilities in the metastate space of size N^K. The cost of the forward or backward calculations is therefore $\mathcal{O}(N^{2K}T)$. As K (the number of feature streams to model, such as

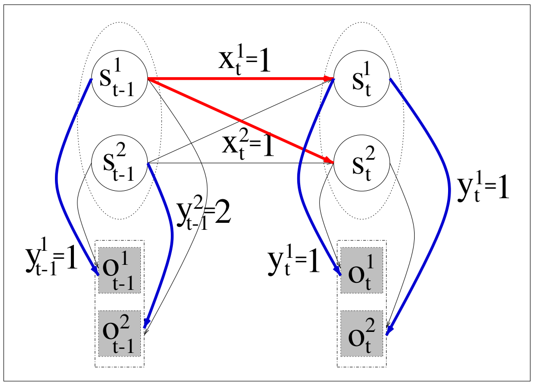

Figure 2: Bold lines show information specified by the hidden (vector) variables Y_{t-1}, X_t, Y_t.

the number of phonological features) or N (the number of states in each chain) increase this becomes intractable. The following sections consider approximate schemes allowing more efficient model estimation and decoding. MCMC-based methods eg. [3]) are also applicable but not investigated here.

3 MORE EFFICIENT FORWARD-BACKWARD ALGORITHM

Forward (and backward) calculations can be made more efficient for the special case of the OMMFHMM, reducing complexity from $\mathcal{O}(N^{2K}T)$ for the standard forward (or backward) algorithm in the metastate space to $\mathcal{O}(KN^{K+1}T)$. The details may be found in [23]. The speedup is similar to that proposed in [8], presented in the context of an alternative parameter reduction scheme for factorial HMMs. Readers familiar with the graphical modelling framework will find it obvious that such a speedup should exist, observing that (1) the cost of the general inference algorithm for graphical models scales as the sum of the sizes of the state spaces of the cliques, and (2) the sum of the sizes of the state spaces of cliques in the observation-only coupled model are smaller than in the fully coupled model [15, 11].

4 ALGORITHMS USING MOST-LIKELY METASTATE SEQUENCE

Large vocabulary ASR systems using HMMs generally assume in recognition that a single state sequence S^* dominates likelihood calculations, ie. $p(\mathbf{O}) \approx p(\mathbf{O}, S^*)$, where if $\mathcal{S}$ denotes the set of length T state sequences through the model then $S^* = \arg\max_{S \in \mathcal{S}} p(S|\mathbf{O})$. The optimal S^* is found using the Viterbi algorithm. The Viterbi algorithm can also be applied directly to the MMFHMM to find the optimal metastate sequence $\mathbf{S}^*$, but for the general MMFHMM the computational order is the same as that of the forward and backward calculations, ie. $\mathcal{O}(N^{2K}T)$[6]. The Chain Viterbi algorithm described below is a more efficient scheme yielding an approximation to $\mathbf{S}^*$. The metastate sequence $\mathbf{S}^*$ can also be used in a *Viterbi training* (or, when $\mathbf{S}^*$ is approximate, a *Viterbi-training-like*) estimation scheme for MMFHMMs: this approach is analogous to the use of standard Viterbi training in HMM systems based on Gaussian mixtures, in which a single state sequence is assumed to dominate but the Gaussian mixtures are trained using the EM algorithm. An alternative, not investigated here, makes the stronger assumption that a single $(\mathbf{S}^*, \mathbf{X}^*, \mathbf{Y}^*)$ sequence dominates likelihood calculations.

4.1 Chain Viterbi Metastate Sequences

A more efficient scheme for *approximating* $\mathbf{S}^*$ when the K time series are *assumed* weakly coupled is proposed in [30][7]. The algorithm iterates through each stream k in turn, finding the optimal sequence of hidden states through stream k given fixed values for the hidden states of the other streams. The state space is thus reduced to size N when doing the optimizations for stream k. The algorithm can be initialized by (for example) computing a Viterbi state sequence for each chain individually or by assuming a uniform segmentation of the observations for each stream. Iteration through all K streams continues until convergence, which is not necessarily to the optimal sequence $\mathbf{S}^*$ (see [24] for a counter-example). Note that softer versions of this iterative scheme would also be possible, using a subroutine where the state sequences corresponding to some subset of streams are fixed and a Viterbi-like decoding procedure applied to the remainder.

5 MEAN-FIELD VARIATIONAL APPROXIMATION

Parameter estimation algorithms based on variational methods are currently popular in the graphical models community, where ML estimation using the EM algorithm is often intractable[8]. This section outlines the basic arguments in the context of the OMMFHMM; [12] is a more general presentation.

Variational methods exploit one particular lower bound on the log likelihood of a set of data for likelihood approximation and as the basis of a parameter estimation procedure. For the specific case of the OMMFHMM, where hidden variables $\mathbf{S}$ and $\mathbf{Y}$ specify the metastate sequence and the specific

[6]Again, for the special case of the OMFHMM, speedups similar to those used by the modified forward or backward algorithms mentioned in Section 3 are possible.

[7]The approach is similar to the Iterative Conditional Modes algorithm for computing a MAP estimate of Markov Random Field parameters [2].

[8]Inference in the general case is NP-hard [5].

observation distributions used at each t, this lower bound is:

$$\begin{aligned}\mathcal{L}(\lambda) = \ln p(\mathbf{O}|\lambda) &= \ln\{\sum_{\mathbf{S},\mathbf{Y}} p(\mathbf{O},\mathbf{S},\mathbf{Y}|\lambda)\} \\ &= \ln\{\sum_{\mathbf{S},\mathbf{Y}} Q(\mathbf{S},\mathbf{Y}|\Psi)\frac{p(\mathbf{O},\mathbf{S},\mathbf{Y}|\lambda)}{Q(\mathbf{S},\mathbf{Y}|\Psi)}\} \\ &\geq \sum_{\mathbf{S},\mathbf{Y}} Q(\mathbf{S},\mathbf{Y}|\Psi)\ln\frac{p(\mathbf{O},\mathbf{S},\mathbf{Y}|\lambda)}{Q(\mathbf{S},\mathbf{Y}|\Psi)} = \mathcal{L}_Q(\Psi,\lambda)\end{aligned}$$

where λ denotes OMFHMM parameters, $\mathcal{L}(\lambda)$ denotes the likelihood function, $Q(\mathbf{S},\mathbf{Y}|\Psi)$ is some distribution over the hidden variables which has parameters Ψ and $\mathcal{L}_Q(\Psi,\lambda)$ denotes the lower bound of interest. The third line follows by Jensen's inequality [28]. Note $\mathcal{L}(\lambda)$ exceeds $\mathcal{L}_Q(\Psi,\lambda)$ by exactly $KL[Q(\mathbf{S},\mathbf{Y}|\Psi)||p(\mathbf{S},\mathbf{Y}|\mathbf{O},\lambda)]$, the Kullback-Leibler (KL) divergence between the distributions $Q(\mathbf{S},\mathbf{Y}|\Psi)$ and $p(\mathbf{S},\mathbf{Y}|\mathbf{O},\lambda)$, which is a non-negative quantity [6]. The lower bound may be tightened for each observation sequence $\mathbf{O}$ by adjusting the *variational parameters* Ψ of the *variational distribution Q* to minimize the KL divergence.

This bound may be useful where likelihood calculations are intractable; [21] observes that the lower bound $\mathcal{L}_Q(\Psi,\lambda)$ may also be usefully applied to ML parameter estimation problems using the following coordinate ascent procedure. Denoting parameters of model p and of variational distribution Q at iteration k by λ^k and Ψ^k respectively:

Step 1 Maximize $\mathcal{L}_Q(\Psi,\lambda^k)$ wrt Ψ. Or, equivalently, minimize $KL[Q(\mathbf{S},\mathbf{Y}|\Psi)||p(\mathbf{S},\mathbf{Y}|\mathbf{O},\lambda^k)]$ wrt Ψ;

Step 2 Maximize $\mathcal{L}_Q(\Psi^{k+1},\lambda)$ wrt λ.

The algorithm is guaranteed to increase the lower bound on the likelihood $\mathcal{L}_Q(\Psi,\lambda)$ at each step, although not necessarily the likelihood $\mathcal{L}(\lambda)$. Convergence of the algorithm may be assessed by monitoring changes in the lower bound.

To examine the operation of the variational learning algorithm, consider first the case where $Q(\mathbf{S},\mathbf{Y}|\Psi)$ is allowed to range over all possible distributions over the hidden variables. In Step 1, the distribution minimizing the KL divergence is $Q(\mathbf{S},\mathbf{Y}|\Psi) = p(\mathbf{S},\mathbf{Y}|\mathbf{O},\lambda^k)$, which results in a KL divergence of zero [6]. Thus, Step 2 seeks

$\lambda^{k+1} = \arg\max_\lambda \sum_{\mathbf{S},\mathbf{Y}} p(\mathbf{S},\mathbf{Y}|\mathbf{O},\lambda^k)\ln p(\mathbf{S},\mathbf{Y}|\mathbf{O},\lambda)$

which is equivalent to the standard EM algorithm [7].

More generally, the bound above is of utility for estimating models where standard EM estimation would be intractable. For these cases, the family of distributions $Q(\mathbf{S},\mathbf{Y}|\Psi)$ is assumed to have a form in which inference is more tractable than in the original model. For example, when working with graphical models, a form for Q allowing more tractable inference is often identified by simplifying the dependencies in the original graphical model p for which inference is intractable. This work adopts a very simple family Q; this *mean-field approximation* is a completely factorized approximation in which all hidden variables are assumed independent given the observations. Thus

$$Q(\mathbf{S}, \mathbf{Y}|\Psi) = \prod_{t=1}^{T} \{ \prod_{k=1}^{K} Q_t^{Sk}(s_t^k|\Psi_t^{Sk}) Q_t^{Yk}(y_t^k|\Psi_t^{Yk}) \}$$

where

- $Q_t^{Sk}(s_t^k|\Psi_t^{Sk})$ denotes a pmf with parameters $\Psi_t^{Sk} = \{\Psi_{tj}^{Sk}|j^k \in \Theta_k\}$ and Ψ_{tj}^{Sk} denotes the probability of a particular outcome j^k;
- $Q_t^{Yk}(y_t^k|\Psi_t^{Yk})$ denotes a pmf with parameter set $\Psi_t^{Yk} = \{\Psi_{tl}^{Yk}|1 \le l \le K\}$, and Ψ_{tl}^{Yk} denotes the probability of a particular outcome l.

To simplify maintenance of positivity, ensure appropriate normalization and guarantee that no hidden event is assigned probability zero, each variational pmf will be assumed to have a softmax form. So for each $j^k \in \Theta_k$

$$Q_t^{Sk}(s_t^k = j^k|\Psi_t^{Sk}) \stackrel{\text{def}}{=} \frac{\exp \Psi_{tj}^{Sk}}{\sum_{i \in \Theta_k} \exp \Psi_{ti}^{Sk}}$$

and for each $1 \le l \le K$

$$Q_t^{Yk}(y_t^k = l|\Psi_t^{Yk}) \stackrel{\text{def}}{=} \frac{\exp \Psi_{tl}^{Yk}}{\sum_{\nu=1}^{K} \exp \Psi_{t\nu}^{Yk}}$$

Minimization of KL divergence (for lower bound calculation or for Step 1 of variational learning) can be implemented using, for example, a basic gradient descent procedure. The updates for model observation-related parameters λ are obtained using a derivation similar to that for the standard M-step of the EM algorithm (eg. [23]). More details may be found in [24]. In this work, updates are made only to the observation-related parameters $\phi^k(l)$ and $b^{kl}(o_t^k|i^l)$; the scheme provides a computationally cheap means of integrating K independent HMMs which have been trained on the K streams individually. Finally, it is noted in passing that more structured variational approximations are possible (see for example [12, 8]) but these typically have higher computational cost.

6 EXPERIMENTAL RESULTS

Experiments use the OGI ISOLET database [4], which comprises wideband recordings of isolated utterances of single letters of the alphabet. Whilst far from the conversational speech motivating the research, use of a simple testbed such as ISOLET facilitates an initial feasibility study of novel models and algorithms without the additional complications introduced by continuous speech tasks. We use *Isolet1-4* (6240 utterances) to train and the speaker-disjoint *Isolet5* (1560 utterances) to test[9]. For reference, performance of our baseline HMMs using a 39-dimensional observation vector of *full-band* cepstra (including 0th) with delta and acceleration coefficients is between 96.2% (3 state HMM) and 96.6% (10 state HMM) for this data set. The experimental task considered here is that of modelling cepstra derived from frequency subbands (eg. [20, 31, 19]), rather than a more speculative articulatory or phonological representation. There is evidence that asynchrony exists between different frequency bands [20], making this a reasonable task for a feasibility study; however, there is likely to be more asynchrony in articulatory or phonological representations, where the advantages of factorial models are expected to be more evident.

The extraction of subband cepstra proceeds as follows. 25ms windows of speech are Fourier-transformed and filtered through a bank of 20 overlapping, equally mel-spaced, filters. Filtering produces a vector of log spectral energies $E = [e_1, \ldots, e_{20}]$. A choice of V frequency subbands subdivides E into V subvectors E_v. A DCT D_v is applied to each E_v to yield a vector of cepstra $C_v = D_v E_v$ for subband v. Decreasing D_v row dimensionality effects cepstral truncation, reducing the dimensionality of C_v from that of E_v: a V-tuple $(\#_1, \ldots, \#_V)$ denotes the truncation scheme, where $\#_v$ indicates retention of cepstra $0, \ldots, \#_v - 1$ in subband v. Finally, observations for the v-th subband stream (o_t^v in our earlier notation) are formed by appending the appropriate delta and acceleration coefficients to C_v. The experiments below use cepstra from two or three subbands. Observations for the two-stream experiments comprise cepstra from two subbands 0-2 and 2-8kHz, with cepstral truncation (7,6), yielding a 39-d combined observation vector $\mathbf{O}_t$. Observations for three-stream experiments comprise cepstra from three subbands 0-0.9, 0.8-2.7, 2.7-8kHz, with cepstral truncation (5,4,4), again yielding a 39-d combined observation vector $\mathbf{O}_t$.

6.1 Comparison: Approximate Decoding Schemes

This section uses OMMFHMM models to study the quality of the likelihood approximations and associated classification performance given by Chain Viterbi metastate sequences and by the Mean-Field variational lower bound. Results obtained using Viterbi Decoding are included as a further baseline for comparison.

6.1.1 Experimental Setup

Each experiment uses the *same*, fixed set of observation coupled models, which were trained using the EM algorithm.

Chain Viterbi Initialization: two schemes were investigated for specifying an initial metastate sequence. The first used a uniform segmentation of the observations in stream k against states in chain k; the second used the segmentation obtained by doing a Viterbi decoding of stream k observations using the chain k-related parameters only (for each k). Preliminary experiments found initialization had little effect on results; results below are for the per-chain Viterbi initialization.

[9] This training set is twice as large as that used by [4, 13]. All experiments train a single, distinct model for each letter; this differs from the HMM-based system of [13] which utilizes parameter sharing across members of the E-set of letters and also uses an explicit silence model.

Mean-Field Initialization and Thresholds: two initializations of the $Q_t^{Sk}(j)$ distributions were investigated. Initial per-stream state sequences were either obtained using the *uniform* or the *per-chain Viterbi decoding* schemes as described for the Chain Viterbi initializations; each $Q_t^{Sk}(j)$ distribution was then initialized with a soft version of this distribution, assigning mass 0.8 to the state occupied in the Viterbi or uniform segmentation, and distributing mass the remaining mass equally amongst the remaining states. $Q_t^{Yk}(l)$ distributions were initialized as a uniform distribution. Preliminary experiments found the per-chain Viterbi initialization gave considerably better results and so this was used to produce the results below. The naive gradient-descent implementation also requires a stepsize: a brute force search over a range of values was used and the results below correspond to the stepsize yielding the highest value for the lower-bound on test set likelihood (*not* the stepsize giving the best *classification performance*, since this would constitute cheating).

6.1.2 Experimental Results

Tables 1 and 2 compare classification %Correct performance when using exact full likelihood, Viterbi and Chain Viterbi approximations and the Mean-Field Variational lower bound with a ML decision rule. The tables also show results of tests for significant differences between full-likelihood classification and the approximate algorithms, calculated using the McNemar test [9]. Figure 3 compares total test set likelihood under each class A-Z with the values obtained from Viterbi and Chain Viterbi approximations and from the Mean-Field Variational lower bound when modelling three subband cepstral streams. This and other graphs obtained all show that the Chain Viterbi likelihood approximation is considerably closer to the exact likelihood than the Mean-Field Variational lower bound. The difference between classification performance using full-likelihood and Chain Viterbi classification is rarely found to be significant; it appears that the initialization used rarely leads to convergence to a suboptimal local maxima. The mean-field approximation is less successful. This is presumably due to the extreme nature of the mean-field assumption: a more costly but more structured variational approximation might be more successful. Appendix A includes approximate execution times for the different algorithms for the case of models with $K = L = 3$.

states per stream	Full Likelihood %C	Viterbi %C	Chain Viterbi %C	Mean Field %C
3	94.9	94.8	95.0	91.0*($p \approx 0$)
6	95.3	95.4	95.3	95.1
8	96.0	96.0	96.0	96.0

Table 1: 2-Stream Results: Decoding Schemes (*=significantly different at $\alpha = 0.01$ level, with corresponding p value in brackets)

6.2 Comparison: Approximate Training and Decoding Schemes

This subsection compares the classification performance of OMMFHMMs trained and tested using *matched* (exact or approximate) algorithms ie. EM training is used with full likelihood (FL) classifica-

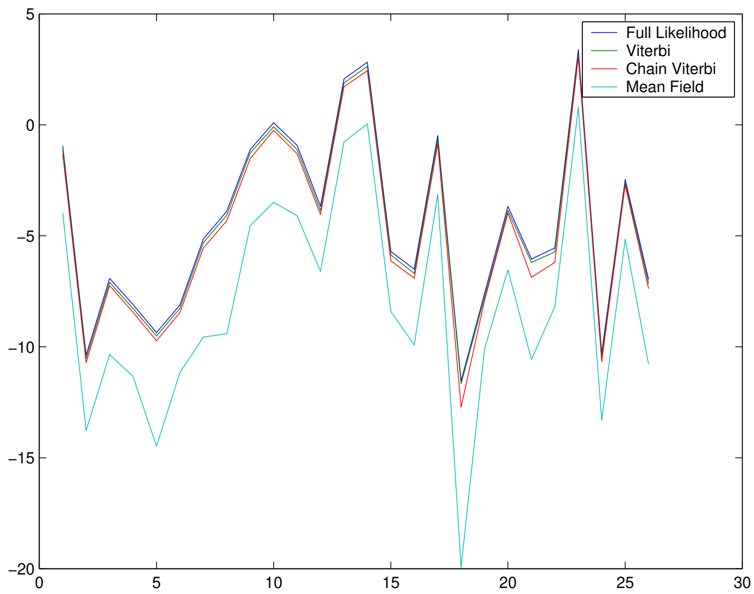

Figure 3: Approximations To Test-Set Likelihoods, 3 streams, 8-state models (X-axis corresponds to classes A-Z, with A=1 through Z=26.)

tion, Chain Viterbi training is used with a Chain Viterbi approximation in classification etc. Results obtained using Viterbi Training and Decoding are included as a further baseline for comparison.

6.2.1 Experimental Setup

All training algorithms stop one iteration after the gain in data likelihood or the variational lower bound drops below 1%.

Chain Viterbi Initialization: the initial metastate sequence was obtained by doing a Viterbi decod-

states per stream	Full Likelihood %C	Viterbi %C	Chain Viterbi %C	Mean Field %C
3	94.9	95.0	96.2	91.7*($p \approx 0$)
6	96.4	96.3	95.0	95.2*($p \approx 0$)
8	96.4	96.3	96.2	95.3*($p = 0.009$)

Table 2: 3-Stream Results: Decoding Schemes (*=significantly different at $\alpha = 0.01$ level, with corresponding p value in brackets)

ing of stream k observations using the chain k-related parameters only, since this proved a useful initialization in the decoding-only experiments of the previous section.

Mean-Field Initialization and Thresholds: initial per-stream state sequences were obtained using the *per-chain Viterbi decoding* schemes as in Chain Viterbi initialization; each $Q_t^{Sk}(j)$ distributions was then initialized with a soft version of this distribution, assigning mass 0.8 to the state occupied in the Viterbi or uniform segmentation, and distributing the remaining mass equally amongst the remaining states. $Q_t^{Yk}(l)$ distributions were initialized as a uniform distribution. The gradient-descent stepsize used in training and decoding was fixed to the value that was most effective in the decoding-only experiments.

6.2.2 Experimental Results

Tables 3 and 4 compare classification %Correct performance obtained using the exact and approximate schemes with a ML decision rule. The tables also show results of tests for significant differences between exact likelihood-based training and classification and the approximate algorithms, using the McNemar test [9].

states per stream	Full Likelihood %C	Viterbi %C	Chain Viterbi %C	Mean Field %C
3	94.9	94.9	94.2*($p = 0.019$)	93.9
6	95.3	95.4	95.2	95.0
8	96.0	95.8	95.9	96.0

Table 3: 2-Stream Results: Matched Training/Decoding Schemes (*=significantly different at $\alpha = 0.01$ level, with corresponding p value in brackets)

states per stream	Full Likelihood %C	Viterbi %C	Chain Viterbi %C	Mean Field %C
3	94.9	95.0	94.9	92.5*($p \approx 0$)
6	96.4	95.8	95.6	95.3
8	96.4	96.3	96.1	95.7

Table 4: 3-Stream Results: Matched Training/Decoding Schemes (*=significantly different at $\alpha = 0.01$ level, with corresponding p value in brackets)

Classification performance achieved when training and decoding using the approximate algorithms is not significantly different to that achieved using the exact scheme in most cases. Results suggest that the EM, Viterbi and Chain Viterbi algorithms all take a similar number of iterations to fall below the relative change training termination threshold; the Mean-Field scheme takes fewer. Appendix A includes approximate execution times for the different algorithms for the case of models with $K = L = 3$.

7 CONCLUSIONS AND FUTURE WORK

The results above suggest that the Chain Viterbi algorithm is an adequate approximate algorithm appropriate for use in decoding, giving likelihood approximations very close to the true values. There is a less marked difference between the two approximate schemes when used in both training and testing; however, since the Chain Viterbi algorithm requires no 'tuning' parameters such as step sizes, it would also be our algorithm of choice for use in future work.

Our current work is investigating an alternative to the mixed-memory approximation for parameter reduction in factorial HMMs. Future work will consider methods for applying these models within continuous speech recognition systems. The primary question still to be addressed relates to the choice of modelling units and thus the degree of allowable asynchrony; other areas meriting investigation include the incorporation of discriminatively-estimated exponent stream weights and the use of alternative representations of speech.

8 ACKNOWLEDGEMENTS

The authors thank Dr Mark Gales, Mr Phil Woodland and members of the SSLI Lab for helpful discussions. This research was partially supported by an EPSRC Advanced Studentship and by DARPA contract No. N660019928924. The views and conclusions contained in this document are those of the authors and should not be interpreted as reflecting the official policies of the funding agency.

References

[1] H Attias. Personal Communication.

[2] J Besag. On The Statistical Analysis Of Dirty Pictures. *Journal of the Royal Statistical Society, Series B*, 48(3):259–302, 1986.

[3] J Besag. Markov Chain Monte Carlo for Statistical Inference. Working Paper 9, Center for Statistics and the Social Sciences, University of Washington, 2000.

[4] R Cole, Y Muthusamy, and M Fanty. The ISOLET spoken letter database. Technical Report CSE 90-004, OGI, 1990.

[5] G Cooper and E Herskovits. Computational Complexity Of Probabilistic Inference Using Bayesian Belief Networks. *Artificial Intelligence*, pages 393–405, 1990.

[6] T Cover and J Thomas. *Elements Of Information Theory*. John Wiley and Sons, 1991.

[7] AP Dempster, NM Laird, and DB Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977.

[8] Z Ghahramani and MI Jordan. Factorial Hidden Markov Models. *Machine Learning*, 29:245–273, 1997.

[9] L Gillick and SJ Cox. Some Statistical Issues In The Comparison Of Speech Recognition Algorithms. In *Proc ICASSP*, pages 532–535, 1989.

[10] D Graff. The 1996 Broadcast News Speech And Language-Model Corpus. In *Proc DARPA Speech Recognition Workshop*, February 1997.

[11] FV Jensen. *An Introduction To Bayesian Networks*. Springer, 1996.

[12] M Jordan, Z Ghahramani, T Jaakkola, and L Saul. An Introduction to Variational Methods for Graphical Models. In *Learning In Graphical Models*, pages 105–161. MIT Press, 1999.

[13] S Kapadia. *Discriminative Training of Hidden Markov Models*. PhD thesis, University of Cambridge Engineering Dept, 1998.

[14] K Kirchhoff. Robust Speech Recognition Using Articulatory Information. Technical Report 98-036, International Computer Science Institute, 1998.

[15] SL Lauritzen. *Graphical Models*. Oxford: Clarendon, 1996.

[16] J Li and A Murua. A 2d Extended HMM For Speech Recognition. In *Proc ICASSP*, pages 349–352, 1999.

[17] BT Logan and PJ Moreno. Factorial HMMs for Acoustic Modeling. In *Proc ICASSP*, pages 813–816, 1998.

[18] S Matsuda, M Nakai, H Shimodaira, and S Sagayama. Asynchronous-Transition HMM. In *Proc ICASSP*, pages 1001–1004, 2000.

[19] P McMahon, P McCourt, and S Vaseghi. Discriminative Weighting of Multi-Resolution Sub-Band Cepstral Features for Speech Recognition. In *Proc ICSLP*, 1998.

[20] N Mirghafori. *A Multi-Band Approach to Automatic Speech Recognition*. PhD thesis, ICSI, UC Berkeley, 1999.

[21] RM Neal and GE Hinton. A New View Of The EM Algorithm That Justifies Incremental And Other Variants. Unpublished Manuscript, 1993.

[22] HJ Nock and SJ Young. Loosely-Coupled HMMs for ASR. In *Proc ICSLP*, Beijing, China, October 2000.

[23] HJ Nock and SJ Young. Loosely Coupled HMMs For ASR: A Preliminary Study. Technical Report CUED/F-INFENG/TR386, CUED, 2000.

[24] HJ Nock and SJ Young. Approximate Algorithms For Training Loosely-Coupled HMMs. Technical Report CUED/F-INFENG/TR409, Cambridge University Engineering Department, 2001.

[25] J Odell. *The Use Of Context In Large Vocabulary Speech Recognition*. PhD thesis, University of Cambridge Engineering Dept, 1995.

[26] M Ostendorf and H Singer. HMM Topology Design Using Maximum Likelihood Successive State Splitting. *Computer Speech And Language*, 11:17–41, 1997.

[27] AE Raftery. A Model For High-Order Markov Chains. *Journal of the Royal Statistical Society, Series B*, 47(3), 1985.

[28] S Ross. *A First Course In Probability*. Prentice Hall, 1997.

[29] M Saraclar, H Nock, and S Khudanpur. Pronunciation Modeling By Sharing Gaussian Densities Across Phonetic Models. *Computer Speech And Language*, 14:137–160, 2000.

[30] LK Saul and MI Jordan. Mixed Memory Markov Models. *Machine Learning*, 37:75–87, 1999.

[31] MJ Tomlinson, MJ Russell, RK Moore, AP Buckland, and MA Fawley. Modelling Asynchrony in Speech Using Elementary Single-Signal Decomposition. In *Proc ICASSP*, pages 1247–1250, 1997.

[32] M Weintraub, K Taussig, K Hunicke-Smith, and A Snodgrass. Effect Of Speaking Style On LVCSR Performance. In *Proc ICSLP*, pages S16–S19 (addendum), Philadephia, USA, 1996.

[33] PC Woodland, CJ Leggetter, JJ Odell, and SJ Young. The Development of the 1994 HTK Large Vocabulary Speech Recognition System. In *Proc ARPA Spoken Language Systems Technology Workshop*, pages 110–115, 1995.

[34] S Young, J Jansen, J Odell, D Ollason, and P Woodland. *The HTK Book (Version 2.0)*. ECRL, 1995.

[35] GG Zweig. *Speech Recognition With Dynamic Bayesian Networks*. PhD thesis, University of California, Berkeley, 1998.

A Timing Comparisons

The timings in Table 5, given in seconds, represent the average time taken to decode the entire test set using the model for a single class (ie. a letter). The timings in Table 6, given in seconds, represent the average time taken to train the model for a single class. The average timings are included to give an impression of general trends, but they are not completely accurate for two reasons. Decoding or training corresponding to a single class was performed separately, using a CPU from amongst a set of dual processor P3 machines with 256k secondary cache, running linux, but with differing processor speeds between 700MHz-1GHz and with bus speeds of 100 or 133 MHz; the set of CPUs used by any one algorithm may differ slightly. Execution time is also affected by the program executing on the second processor and this can change execution times by 10-20%. In addition, few efforts were made to optimize the implementations, so these figures should not be interpreted as the best that is achievable using any of the algorithms. For example, some researchers have found that use of fixed point iteration rather than gradient descent in the implementation of variational algorithms leads to faster convergence [1].

A variety of gradient descent step sizes were investigated when calculating the mean-field variational lower bound for use in decoding. The three timing numbers included here are for the largest, intermediate and smallest step sizes investigated. The gradient descent step-size for mean-field variational training is fixed to that which gave the highest lower bound in decoding-only experiments.

	3 states per chain	6 states per chain	8 states per chain
FL	178	7442	37520
VIT	81	1424	7199
CVIT (Per-Chain Viterbi Initialization)	209	440	713
MF (Per-Chain Viterbi Initialization, ordered by Step Size: Smallest; Intermediate; Largest)	(2005;273;224)	(4938;656;529)	(11235;1424;990)

Table 5: Decoding Algorithms: Comparison of Execution Times (3 stream models)

	3 states per chain	6 states per chain	8 states per chain
EM	690	28485	219488
VIT	97	1247	5501
CVIT (Per-Chain Viterbi Initialization)	86	181	294
MF (Per-Chain Viterbi Initialization)	7574	8373	13716

Table 6: Training Algorithms: Comparison of Execution Times (3 stream models)