Factor analysed hidden Markov models for speech recognition

A-V.I. Rosti *, M.J.F. Gales

Cambridge University Engineering Department, Trumpington Street, Cambridge, CB2 1PZ, UK

Abstract

Recently various techniques to improve the correlation model of feature vector elements in speech recognition systems have been proposed. Such techniques include semi-tied covariance HMMs and systems based on factor analysis. All these schemes have been shown to improve the speech recognition performance without dramatically increasing the number of model parameters compared to standard diagonal covariance Gaussian mixture HMMs. This paper introduces a general form of acoustic model, the factor analysed HMM. A variety of configurations of this model and parameter sharing schemes, some of which correspond to standard systems, were examined. An EM algorithm for the parameter optimisation is presented along with a number of methods to increase the efficiency of training. The performance of FAHMMs on medium to large vocabulary continuous speech recognition tasks was investigated. The experiments show that without elaborate complexity control an equivalent or better performance compared to a standard diagonal covariance Gaussian mixture HMM system can be achieved with considerably fewer parameters.

Key words: Hidden Markov models, state space models, factor analysis, speech recognition

1 Introduction

Hidden Markov models (HMMs) with continuous observation densities have been widely used for speech recognition tasks. The observation densities associated with each state of the HMMs should be sufficiently general to capture

* Corresponding author.
Email addresses: avir2@eng.cam.ac.uk (A-V.I. Rosti), mjfg@eng.cam.ac.uk (M.J.F. Gales).

the variations among individual speakers and acoustic environments. At the same time, the number of parameters describing the densities should be as low as possible to enable fast and robust parameter estimation when using a limited amount of training data. Gaussian mixture models (GMMs) are the most commonly used form of state distribution model. They are able to approximate non-Gaussian densities, including densities with multiple modes. One of the issues when using multivariate Gaussian distributions or GMMs is the form of covariance matrix for each component. Using full covariance components increases the number of parameters dramatically which can result in poor parameter estimates. Hence, components with diagonal covariance matrices are commonly used in HMMs for speech recognition. Diagonal covariance GMMs can approximate correlations between the feature vector elements. However, it would be beneficial to have uncorrelated feature vectors for each component when diagonal covariance matrices are used.

A number of schemes to tackle this intra-frame correlation problem have been proposed. One approach to decorrelate the feature vectors is to transform each set of vectors assigned to a particular component so that the diagonal covariance matrix assumption becomes valid. This system would, however, have the same complexity as full covariance GMMs. Alternatively, a single global decorrelation transform could be used as in principal component analysis. Unfortunately, it is hard to find a single transform that decorrelates speech feature vectors for all states in an HMM system. Semi-tied covariance matrices (STCs) (Gales, 1999) can be viewed as a halfway solution. A class of states with diagonal covariance matrices can be transformed into full covariance matrices via a class specific linear transform. Systems employing STC generally yield better performance than standard diagonal covariance HMMs, or single global transforms, without dramatically increasing the number of model parameters.

The intra-frame correlation modelling problem may also be addressed by using subspace models. Heteroscedastic linear discriminant analysis (HLDA) (Gales, 2002; Kumar, 1997) models the feature vectors via a linear projection matrix applied to some lower dimensional vectors superimposed with noise spanning the uninformative, “nuisance” dimensions. There is a close relationship between STC and HLDA. The parameter estimation is similar and both can be viewed as feature space transform schemes. Alternatives to systems based on LDA-like projections are schemes based on factor analysis (Saul and Rahim, 1999; Gopinath, Ramabhadran, and Dharanipragada, 1998). These model the covariance matrix via a linear probabilistic process applied to a simpler lower dimensional representation called factors. Where the LDA can be viewed as a projection scheme the factor analysis is later referred to as a linear transformation due to the additive noise term. The factors can be viewed as state vectors and the factor analysis as a generative observation process. Each component of a standard HMM system can be replaced with a factor analysed covariance model (Saul and Rahim, 1999). This dramatically increases the number of model parameters due to an individual loading matrix attached to each component. The loading matrix (later referred to as the observation matrix) and the underlying factors (state vector) can be shared among several observation noise components as in shared factor analysis (SFA). This system is closely related to the "factor analysis invariant to linear transformations of data" (FACILT) (Gopinath et al., 1998) without the global linear transformation. SFA also assumes the factors being distributed according to a standard normal distribution. Alternatively the standard factor analysis can be extended by modelling the factors with GMMs as in independent factor analysis (IFA) (Attias, 1999). In IFA the individual factors are modelled by independent 1-dimensional GMMs.

This paper introduces an extension to the standard factor analysis which is applicable to HMMs. The model is called factor analysed HMM (FAHMM). FAHMMs belong to a broad class of generalised linear Gaussian models (Rosti and Gales, 2001) which extends the set of standard linear Gaussian models (Roweis and Ghahramani, 1999). Generalised linear Gaussian models are state space models with linear state evolution and observation processes, and Gaussian mixture distributed noise processes. The underlying HMM generates piecewise constant state vector trajectories that are mapped into the observation space via linear probabilistic observation processes. FAHMM combines the observation process from SFA with the standard diagonal covariance Gaussian mixture HMM acting as a state evolution process. Alternatively, it can be viewed as a dynamic version of IFA [1] with a Gaussian mixture model as the observation noise. Due to the factor analysis based observation process, FAHMMs should model the intra-frame correlation better than diagonal covariance matrix HMMs, yet be more compact than full covariance matrix HMMs. In addition, FAHMMs allow a variety of configurations and subspaces to be explored.

The model complexity has become a standard problem in speech recognition and machine learning over the recent years (Liu, Gales, and Woodland, 2003). For example, Bayesian information criterion has been applied separately to speaker clustering and selecting the number of Gaussian mixture components in (Chen and Gopalakrishnan, 1998). Current complexity controls are derived from Bayesian schemes based on correctly modelling some held-out data. However, it is well known that the models giving highest log-likelihood for some data do not automatically result in better recognition performance on unseen data. Most of the complexity control work for speech recognition has addressed

[1] The independent factor assumption in IFA would correspond to a multiple stream HMM with independent 1-dimensional streams in the state space. In FAHMMs this assumption is relaxed, and the factors are distributed according to a GMM with diagonal covariance matrices.

the selection of a single form of parameter such as the number of Gaussian components. To date, a successful scheme to select more than one form of parameter simultaneously has not been published. In case of FAHMMs, the number of Gaussian components in both, the state and observation space, as well as the dimensionality of the state space can be chosen. Although the model complexity is an important issue with FAHMMs, it is beyond the scope of this article.

The second section of this paper describes the theory behind FAHMMs including efficient likelihood calculation and the parameter estimation. Implementation issues arising from increased number of model parameters and resource constraints are discussed in the following section. An efficient two level training scheme is described as well. Three sets of experiments with different configurations in medium to large vocabulary speech recognition tasks are presented in Section 4. Conclusions and future work are also provided.

1.1 Notation

In this paper, bold capital letters are used to denote matrices, e.g. $\boldsymbol{A}$, bold letters refer to vectors, e.g. $\boldsymbol{a}$, and plain letters represent scalars, e.g. c. All vectors are column vectors unless otherwise stated. Prime is used to denote the transpose of a matrix or a vector, e.g. $\boldsymbol{A}', \boldsymbol{a}'$. The determinant of a matrix is denoted by $|\boldsymbol{A}|$. Gaussian distributed vectors, e.g. $\boldsymbol{x}$ with mean vector, $\boldsymbol{\mu}$, and covariance matrix, $\boldsymbol{\Sigma}$, are denoted by $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. The likelihood of a vector $\boldsymbol{z}$ being generated by the above Gaussian; i.e., the Gaussian evaluated at the point $\boldsymbol{z}$, is represented as $p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$. Vectors distributed according to a Gaussian mixture model are denoted by $\boldsymbol{x} \sim \sum_m c_m \mathcal{N}(\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)$ where c_m are the mixture weights, and sum to unity. The lower case letter p is used to represent a continuous distribution, whereas a capital letter P is used to denote a probability mass function of a discrete variable. The probability that a discrete random variable, ω, equals m is denoted by $P(\omega = m)$.

2 Factor Analysed Hidden Markov Models

First, the theory behind factor analysis is revisited and a generalisation of factor analysis to encompass Gaussian mixture distributions is presented. The factor analysed HMM is introduced in a generative model framework. Efficient likelihood calculation and parameter optimisation for FAHMMs are then presented. The section is concluded by relating several configurations of FAHMMs to standard systems.

Factor analysis is a statistical method for modelling the covariance structure of high dimensional data using a small number of latent (hidden) variables. It is often used to model the data instead of a Gaussian distribution with full covariance matrix. Factor analysis can be described by the following generative model

$$\boxed{\begin{aligned} &\boldsymbol{x} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}) \\ &\boldsymbol{o} = \boldsymbol{C}\boldsymbol{x} + \boldsymbol{v}, \qquad \boldsymbol{v} \sim \mathcal{N}(\boldsymbol{\mu}^{(o)}, \boldsymbol{\Sigma}^{(o)}) \end{aligned}}$$

where $\boldsymbol{x}$ is a collection of k factors (k-dimensional state vector) and $\boldsymbol{o}$ is a p-dimensional observation vector. The covariance structure is captured by the factor loading matrix (observation matrix), $\boldsymbol{C}$, which represents the linear transform relationship between the state vector and the observation vector. The mean of the observations is determined by the error (observation noise) modelled as a single Gaussian with mean vector $\boldsymbol{\mu}^{(o)}$ and diagonal covariance matrix $\boldsymbol{\Sigma}^{(o)}$. The observation process can be expressed as a conditional distribution, $p(\boldsymbol{o}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{o}; \boldsymbol{C}\boldsymbol{x} + \boldsymbol{\mu}^{(o)}, \boldsymbol{\Sigma}^{(o)})$. Also, the observation distribution is a Gaussian with mean vector $\boldsymbol{\mu}^{(o)}$ and covariance matrix $\boldsymbol{C}\boldsymbol{C}' + \boldsymbol{\Sigma}^{(o)}$.

The number of model parameters in a factor analysis model is $\eta = p(k+2)$. It should be noted that any non-zero state space mean vector, $\boldsymbol{\mu}^{(x)}$, can be absorbed by the observation mean vector by adding $\boldsymbol{C}\boldsymbol{\mu}^{(x)}$ into $\boldsymbol{\mu}^{(o)}$. Furthermore, any non-identity state space covariance matrix, $\boldsymbol{\Sigma}^{(x)}$, can be transformed into an identity matrix using eigen decomposition, $\boldsymbol{\Sigma}^{(x)} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}'$. The matrix $\boldsymbol{Q}$ consists of the eigenvectors of $\boldsymbol{\Sigma}^{(x)}$ and $\boldsymbol{\Lambda}$ is a diagonal matrix with the eigenvalues of $\boldsymbol{\Sigma}^{(x)}$ on the main diagonal. The eigen decomposition always exists and is real valued since a valid covariance matrix is symmetric and positive definite. The transformation can be subsumed into the observation matrix by multiplying $\boldsymbol{C}$ from the right by $\boldsymbol{Q}\boldsymbol{\Lambda}^{1/2}$. It is also essential that the observation noise covariance matrix be diagonal. Otherwise, the sample statistics of the data can be set as the observation noise and the loading matrix equal to zero. As the number of model parameters in a Gaussian with full covariance matrix is $\eta = p(p+3)/2$, a reduction in the number of model parameters using factor analysis model can be achieved by choosing the state space dimensionality according to $k < (p-1)/2$.

Factor analysis has been extended to employ Gaussian mixture distributions for the factors in IFA (Attias, 1999) and the observation noise in SFA (Gopinath et al., 1998). As in the standard factor analysis above, there is a degeneracy present in these systems. The covariance matrix of one state space component can be subsumed into the loading matrix and one state space noise mean vector can be absorbed by the observation noise mean. Therefore, the factors in SFA can be assumed to obey standard normal distribution. The effective number of free parameters (mixture weights not included) in a factor analysis model with Gaussian mixture noise models is given by $2(M^{(x)}-1)k + kp + 2M^{(o)}p$ where $M^{(x)}$ and $M^{(o)}$ represent the number of mixture components in the state and observation space respectively.

2.2 Generative Model of Factor Analysed HMM

Factor analysed hidden Markov model is a dynamic state space generalisation of a multiple component factor analysis system. The k-dimensional state vectors, $\boldsymbol{x}_t$, are generated by a standard diagonal covariance Gaussian mixture HMM. The p-dimensional observation vectors, $\boldsymbol{o}_t$, are generated by a multiple noise component factor analysis observation process. A generative model for FAHMM can be described by the following equation

$$\boxed{\begin{aligned} \boldsymbol{x}_t &\sim \mathcal{M}^{hmm}, & \mathcal{M}^{hmm} &= \{a_{ij}, c_{jn}^{(x)}, \boldsymbol{\mu}_{jn}^{(x)}, \boldsymbol{\Sigma}_{jn}^{(x)}\} \\ \boldsymbol{o}_t &= \boldsymbol{C}_t\boldsymbol{x}_t + \boldsymbol{v}_t, & \boldsymbol{v}_t &\sim \sum_m c_{jm}^{(o)}\mathcal{N}(\boldsymbol{\mu}_{jm}^{(o)}, \boldsymbol{\Sigma}_{jm}^{(o)}) \end{aligned}} \tag{1}$$

where the observation matrices, $\boldsymbol{C}_t$, may be dependent on the HMM state or tied over multiple states. The HMM state transition probabilities from state i to state j are represented by a_{ij} and the state and observation space mixture distributions are described by the mixture weights $\{c_{jn}^{(x)}, c_{jm}^{(o)}\}$, mean vectors $\{\boldsymbol{\mu}_{jn}^{(x)}, \boldsymbol{\mu}_{jm}^{(o)}\}$ and diagonal covariance matrices $\{\boldsymbol{\Sigma}_{jn}^{(x)}, \boldsymbol{\Sigma}_{jm}^{(o)}\}$.

Dynamic Bayesian networks (DBN) (Ghahramani, 1998) are often presented in conjunction with the generative models to illustrate the conditional independence assumptions made in a statistical model. A DBN describing a FAHMM is shown in Fig. 1. The square nodes represent discrete random variables such as the HMM state $\{q_t\}$, and $\{\omega_t^x, \omega_t^o\}$ which indicate the active state and observation mixture components, respectively. Continuous random variables such as the state vectors, $\boldsymbol{x}_t$, are represented by round nodes. Shaded nodes depict observable variables, $\boldsymbol{o}_t$, leaving all the other FAHMM variables hidden. A conditional independence assumption is made between variables that are not connected by directed arcs. The state conditional independence assumption between the output distributions of a standard HMM is also used in a FAHMM.

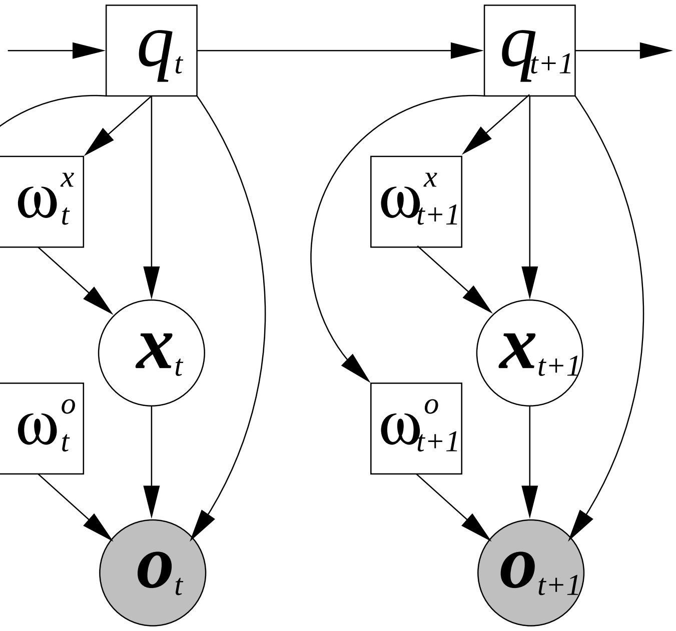

Fig. 1. Dynamic Bayesian network representing a factor analysed hidden Markov model. Square nodes represent discrete and round nodes continuous random variables. Shaded nodes are observable; all the other nodes are hidden. A conditional independence assumption is made between nodes that are not connected by directed arcs.

2.3 FAHMM Likelihood Calculation

An important aspect of any generative model is the complexity of the likelihood calculations. The generative model in Equation (1) can be expressed by the following two Gaussian distributions

$$p(\boldsymbol{x}_t | q_t = j, \omega_t^x = n) = \mathcal{N}(\boldsymbol{x}_t; \boldsymbol{\mu}_{jn}^{(x)}, \boldsymbol{\Sigma}_{jn}^{(x)}) \tag{2}$$

$$p(\boldsymbol{o}_t | \boldsymbol{x}_t, q_t = j, \omega_t^o = m) = \mathcal{N}(\boldsymbol{o}_t; \boldsymbol{C}_j \boldsymbol{x}_t + \boldsymbol{\mu}_{jm}^{(o)}, \boldsymbol{\Sigma}_{jm}^{(o)}) \tag{3}$$

The distribution of an observation $\boldsymbol{o}_t$ given the state $q_t = j$, state space component $\omega_t^x = n$ and observation noise component $\omega_t^o = m$ can be obtained by integrating the state vector $\boldsymbol{x}_t$ out of the product of the above Gaussians. The resulting distribution is also a Gaussian and can be written as

$$b_{jmn}(\boldsymbol{o}_t) = p(\boldsymbol{o}_t | q_t = j, \omega_t^o = m, \omega_t^x = n) = \mathcal{N}\Big(\boldsymbol{o}_t; \boldsymbol{\mu}_{jmn}, \boldsymbol{\Sigma}_{jmn}\Big) \tag{4}$$

where

$$\boldsymbol{\mu}_{jmn} = \boldsymbol{C}_j \boldsymbol{\mu}_{jn}^{(x)} + \boldsymbol{\mu}_{jm}^{(o)} \tag{5}$$

$$\boldsymbol{\Sigma}_{jmn} = \boldsymbol{C}_j \boldsymbol{\Sigma}_{jn}^{(x)} \boldsymbol{C}_j' + \boldsymbol{\Sigma}_{jm}^{(o)} \tag{6}$$

The state distribution of a FAHMM state j can be viewed as an $M^{(o)}M^{(x)}$ component full covariance matrix GMM with mean vectors given by Equation (5) and covariance matrices given by Equation (6).

The likelihood calculation requires inverting $M^{(o)}M^{(x)}$ full p by p covariance matrices in Equation (6). If the amount of memory is not an issue, the inverses and the corresponding determinants for all the discrete states in the system can be computed prior to starting off with the training and recognition. However, this can rapidly become impractical for a large system. A more memory efficient implementation requires the computation of the inverses and determinants on the fly. These can be efficiently obtained using the following equality for matrix inverses (Harville, 1997)

$$\begin{aligned}&(\boldsymbol{C}_j\boldsymbol{\Sigma}_{jn}^{(x)}\boldsymbol{C}_j' + \boldsymbol{\Sigma}_{jm}^{(o)})^{-1} = \\ &\boldsymbol{\Sigma}_{jm}^{(o)-1} - \boldsymbol{\Sigma}_{jm}^{(o)-1}\boldsymbol{C}_j(\boldsymbol{C}_j'\boldsymbol{\Sigma}_{jm}^{(o)-1}\boldsymbol{C}_j + \boldsymbol{\Sigma}_{jn}^{(x)-1})^{-1}\boldsymbol{C}_j'\boldsymbol{\Sigma}_{jm}^{(o)-1}\end{aligned} \tag{7}$$

where the inverses of the covariance matrices $\boldsymbol{\Sigma}_{jm}^{(o)}$ and $\boldsymbol{\Sigma}_{jn}^{(x)}$ are trivial to compute since they are diagonal. The full matrices, $\boldsymbol{C}_j'\boldsymbol{\Sigma}_{jm}^{(o)-1}\boldsymbol{C}_j + \boldsymbol{\Sigma}_{jn}^{(x)-1}$, to be inverted are only k by k matrices. This is dramatically faster than inverting full p by p matrices if $k \ll p$. The determinants needed in the likelihood calculations can be obtained using the following equality (Harville, 1997)

$$|\boldsymbol{C}_j\boldsymbol{\Sigma}_{jn}^{(x)}\boldsymbol{C}_j' + \boldsymbol{\Sigma}_{jm}^{(o)}| = |\boldsymbol{\Sigma}_{jm}^{(o)}||\boldsymbol{\Sigma}_{jn}^{(x)}||\boldsymbol{C}_j'\boldsymbol{\Sigma}_{jm}^{(o)-1}\boldsymbol{C}_j + \boldsymbol{\Sigma}_{jn}^{(x)-1}|$$

where again the determinants of the diagonal covariance matrices are trivial to compute and often the determinant of the k by k matrix is obtained as a by-product of its inverse; e.g., when using Cholesky decomposition. In a large system, a compromise has to be made between precomputing of the inverse matrices and computing them on the fly. For example, caching of the inverses can be employed because some components are likely to be computed more often than others when pruning is used.

The Viterbi algorithm (Viterbi, 1967) can be used to produce the most likely state sequence the same way as with standard HMMs. The likelihood of an observation $\boldsymbol{o}_t$ given only the state $q_t = j$ can be obtained by marginalising the likelihood in Equation (4) as follows

$$b_j(\boldsymbol{o}_t) = p(\boldsymbol{o}_t|q_t = j) = \sum_{m=1}^{M^{(o)}} c_{jm}^{(o)} \sum_{n=1}^{M^{(x)}} c_{jn}^{(x)} b_{jmn}(\boldsymbol{o}_t) \tag{8}$$

Any Viterbi algorithm based decoder such as *token passing* algorithm (Young, Kershaw, Odell, Ollason, Valtchev, and Woodland, 2000) can be easily modified to support FAHMMs this way. The modifications to forward-backward algorithm are discussed in the following training section.

2.4 Optimising FAHMM Parameters

A maximum likelihood (ML) criterion is used to optimise the FAHMM parameters. It is also possible to find a discriminative training scheme such as minimum classification error (Saul and Rahim, 1999) but for this initial work only ML training is considered. In common with standard HMM training the expectation maximisation (EM) algorithm is used. The auxiliary function for FAHMMs can be written as

$$\mathcal{Q}(\mathcal{M},\hat{\mathcal{M}}) = \sum_{\{Q_T\}} \int P(Q|\boldsymbol{O},\mathcal{M}) p(\boldsymbol{X}|\boldsymbol{O},Q,\mathcal{M}) \log p(\boldsymbol{O},\boldsymbol{X},Q|\hat{\mathcal{M}}) d\{\boldsymbol{X}_T\} \tag{9}$$

where $\{Q_T\}$ and $\{X_T\}$ represent all the possible discrete state and continuous state sequences of length T, respectively. A sequence of observation vectors is denoted by $\boldsymbol{O} = \boldsymbol{o}_1,\dots,\boldsymbol{o}_T$, and $\boldsymbol{X} = \boldsymbol{x}_1,\dots,\boldsymbol{x}_T$ is a sequence of state vectors. The set of current model parameters is represented by $\mathcal{M}$, and $\hat{\mathcal{M}}$ is a set of new model parameters.

The sufficient statistics of the first term, $P(Q|\boldsymbol{O},\mathcal{M})$, in the auxiliary function in Equation (9) can be obtained using the standard forward-backward algorithm with likelihoods given by Equation (8). For the state transition probability optimisation, two sets of sufficient statistics are needed, the posterior probabilities of being in state j at time t, $\gamma_j(t) = P(q_t = j|\boldsymbol{O},\mathcal{M})$, and being in state i at time $t-1$ and in state j at time t, $\xi_{ij}(t) = P(q_{t-1} = i, q_t = j|\boldsymbol{O},\mathcal{M})$. For the distribution parameter optimisation the component posteriors, $\gamma_{jmn}(t) = P(q_t = j, \omega_t^o = m, \omega_t^x = n|\boldsymbol{O},\mathcal{M})$, have to be estimated. These can be obtained within the forward-backward algorithm as follows

$$\gamma_{jmn}(t) = \frac{1}{p(\boldsymbol{O})} c_{jm}^{(o)} c_{jn}^{(x)} b_{jmn}(\boldsymbol{o}_t) \sum_{i=1}^{N_s} a_{ij}\alpha_i(t-1)\beta_j(t)$$

where N_s is the number of HMM states in the model, $\alpha_i(t-1)$ is the standard forward variable representing the joint likelihood of being in state i at time $t-1$ and the partial observation sequence up to $t-1$, $p(q_{t-1} = i, \boldsymbol{o}_1,\dots,\boldsymbol{o}_{t-1})$, and $\beta_j(t)$ is the standard backward variable corresponding to the posterior of the partial observation sequence from time $t+1$ to T given being in state j at time t, $p(\boldsymbol{o}_{t+1},\dots,\boldsymbol{o}_T|q_t = j)$.

The second term, $p(\boldsymbol{X}|\boldsymbol{O}, Q, \mathcal{M})$, in the auxiliary function in Equation (9) is the state vector distribution given the observation sequence and the discrete state sequence. Only the first and second-order statistics are required since the distributions are conditionally Gaussian given the state and the mixture components. Using the conditional independence assumptions made in the model, the posterior can be expressed as

$$p(\boldsymbol{x}_t|\boldsymbol{o}_t, q_t = j, \omega_t^o = m, \omega_t^x = n) = \frac{p(\boldsymbol{o}_t, \boldsymbol{x}_t|q_t = j, \omega_t^o = m, \omega_t^x = n)}{p(\boldsymbol{o}_t|q_t = j, \omega_t^o = m, \omega_t^x = n)}$$

which using Equations (2), (3) and (4) simplifies to a Gaussian distribution with mean vector, $\hat{\boldsymbol{x}}_{jmn}(t)$, and correlation matrix, $\hat{\boldsymbol{R}}_{jmn}(t)$, defined by

$$\hat{\boldsymbol{x}}_{jmn}(t) = \boldsymbol{\mu}_{jn}^{(x)} + \boldsymbol{K}_{jmn}\Big(\boldsymbol{o}_t - \boldsymbol{C}_j\boldsymbol{\mu}_{jn}^{(x)} - \boldsymbol{\mu}_{jm}^{(o)}\Big) \tag{10}$$

$$\hat{\boldsymbol{R}}_{jmn}(t) = \boldsymbol{\Sigma}_{jn}^{(x)} - \boldsymbol{K}_{jmn}\boldsymbol{C}_j\boldsymbol{\Sigma}_{jn}^{(x)} + \hat{\boldsymbol{x}}_{jmn}(t)\hat{\boldsymbol{x}}'_{jmn}(t) \tag{11}$$

where $\boldsymbol{K}_{jmn} = \boldsymbol{\Sigma}_{jn}^{(x)}\boldsymbol{C}'_j\Big(\boldsymbol{C}_j\boldsymbol{\Sigma}_{jn}^{(x)}\boldsymbol{C}'_j + \boldsymbol{\Sigma}_{jm}^{(o)}\Big)^{-1}$. It should be noted that the matrix inverted in the equation for $\boldsymbol{K}_{jmn}$ is the inverse covariance matrix in Equation (7) and the same efficient algorithms presented in Section 2.3 apply.

Given the two sets of sufficient statistics above the model parameters can be optimised by solving a standard maximisation problem. The parameter update formulae for the underlying HMM parameters in FAHMM are very similar to those for the standard HMM (Young et al., 2000) except the above state vector distribution statistics replace the observation sample moments. Omitting the state probabilities, the state space parameter update formulae can be written as

$$\hat{c}_{jn}^x = \frac{\sum_{t=1}^{T}\sum_{m=1}^{M^{(o)}}\gamma_{jmn}(t)}{\sum_{t=1}^{T}\gamma_j(t)} \tag{12}$$

$$\hat{\boldsymbol{\mu}}_{jn}^{(x)} = \frac{\sum_{t=1}^{T}\sum_{m=1}^{M^{(o)}}\gamma_{jmn}(t)\hat{\boldsymbol{x}}_{jmn}(t)}{\sum_{t=1}^{T}\sum_{m=1}^{M^{(o)}}\gamma_{jmn}(t)} \tag{13}$$

$$\hat{\boldsymbol{\Sigma}}^{(x)}_{jn} = \mathrm{diag}\left(\frac{\sum_{t=1}^{T}\sum_{m=1}^{M^{(o)}} \gamma_{jmn}(t)\hat{\boldsymbol{R}}_{jmn}(t)}{\sum_{t=1}^{T}\sum_{m=1}^{M^{(o)}} \gamma_{jmn}(t)} - \hat{\boldsymbol{\mu}}^{(x)}_{jn}\hat{\boldsymbol{\mu}}^{(x)\prime}_{jn}\right) \tag{14}$$

where $\mathrm{diag}(\cdot)$ sets all the off-diagonal elements of the matrix argument to zeros. The cross terms including the new state space mean vectors and the first-order accumulates have been simplified in Equation (14). This can only be done if the mean vectors are updated during the same iteration, and the covariance matrices and the mean vectors are tied on the same level. The parameter tying is further discussed in Section 3.2.

The new observation matrix, $\hat{\boldsymbol{C}}_j$, has to be optimised row by row as in SFA (Gopinath et al., 1998). The scheme adopted in this paper follows closely the maximum likelihood linear regression (MLLR) transform matrix optimisation (Gales, 1998). The lth row vector $\hat{\boldsymbol{c}}_{jl}$ of the new observation matrix can be written as

$$\hat{\boldsymbol{c}}_{jl} = \boldsymbol{k}'_{jl}\boldsymbol{G}^{-1}_{jl}$$

where the k by k matrices $\boldsymbol{G}_{jl}$ and the k dimensional column vectors $\boldsymbol{k}_{jl}$ are defined as follows

$$\boldsymbol{G}_{jl} = \sum_{m=1}^{M^{(o)}} \frac{1}{\sigma^{(o)2}_{jml}} \sum_{t=1}^{T}\sum_{n=1}^{M^{(x)}} \gamma_{jmn}(t)\hat{\boldsymbol{R}}_{jmn}(t)$$
$$\boldsymbol{k}_{jl} = \sum_{m=1}^{M^{(o)}} \frac{1}{\sigma^{(o)2}_{jml}} \sum_{t=1}^{T}\sum_{n=1}^{M^{(x)}} \gamma_{jmn}(t)\left(o_{tl} - \mu^{(o)}_{jml}\right)\hat{\boldsymbol{x}}_{jmn}(t)$$

where $\sigma^{(o)2}_{jml}$ is the lth diagonal element of the observation covariance matrix $\boldsymbol{\Sigma}^{(o)}_{jm}$, o_{tl} and $\mu^{(o)}_{jml}$ are the lth elements of the current observation and the observation noise mean vectors, respectively.

Given the new observation matrix, the observation noise parameters can be optimised using the following formulae

$$\hat{c}^{(o)}_{jm} = \frac{\sum_{t=1}^{T}\sum_{n=1}^{M^{(x)}} \gamma_{jmn}(t)}{\sum_{t=1}^{T} \gamma_j(t)}$$

$$\hat{\boldsymbol{\mu}}^{(o)}_{jm} = \frac{\sum_{t=1}^{T} \sum_{n=1}^{M^{(x)}} \gamma_{jmn}(t)\Big(\boldsymbol{o}_t - \hat{\boldsymbol{C}}_j \hat{\boldsymbol{x}}_{jmn}(t)\Big)}{\sum_{t=1}^{T} \sum_{n=1}^{M^{(x)}} \gamma_{jmn}(t)}$$

$$\begin{aligned}
\hat{\boldsymbol{\Sigma}}^{(o)}_{jm} = & \frac{1}{\sum_{t=1}^{T} \sum_{n=1}^{M^{(x)}} \gamma_{jmn}(t)} \sum_{t=1}^{T} \sum_{n=1}^{M^{(x)}} \gamma_{jmn}(t) \mathrm{diag}\Big(\boldsymbol{o}_t \boldsymbol{o}_t' \\
& - \Big[\hat{\boldsymbol{C}}_j \ \hat{\boldsymbol{\mu}}^{(o)}_{jm}\Big] \Big[\boldsymbol{o}_t \hat{\boldsymbol{x}}'_{jmn}(t) \ \boldsymbol{o}_t\Big]' - \Big[\boldsymbol{o}_t \hat{\boldsymbol{x}}'_{jmn}(t) \ \boldsymbol{o}_t\Big] \Big[\hat{\boldsymbol{C}}_j \ \hat{\boldsymbol{\mu}}^{(o)}_{jm}\Big]' \\
& + \Big[\hat{\boldsymbol{C}}_j \ \hat{\boldsymbol{\mu}}^{(o)}_{jm}\Big] \begin{bmatrix} \hat{\boldsymbol{R}}_{jmn}(t) & \hat{\boldsymbol{x}}_{jmn}(t) \\ \hat{\boldsymbol{x}}'_{jmn}(t) & 1 \end{bmatrix} \Big[\hat{\boldsymbol{C}}_j \ \hat{\boldsymbol{\mu}}^{(o)}_{jm}\Big]' \Big)
\end{aligned} \tag{15}$$

Detailed derivation of the parameter optimisation can be found in (Rosti and Gales, 2001).

A direct implementation of the training algorithm is inefficient due to the heavy matrix computations required to obtain the state vector statistics. An efficient two level implementation of the training algorithm is presented in Section 3.4. Obviously, there is no need to compute the off-diagonal elements of the new covariance matrices in Equations (14) and (15).

2.5 Standard Systems Related to FAHMMs

A number of standard systems can be related to FAHMMs. Since the FAHMM training algorithm described above is based on EM algorithm, it is only applicable if there is observation noise. Some of the related systems have the observation noise set to zero which means that different optimisation methods have to be used. The related systems are presented in Table 1 and their properties are further discussed below.

- By setting the number of state space mixture components to zero, $M^{(x)} = 0$, FAHMM reduces to a standard diagonal covariance Gaussian mixture HMM. The observation noise acts as the state conditional output distribution of the HMM, and the observation matrix is made redundant because no state vectors will be generated.
- By setting the number of state space mixture components to one, $M^{(x)} = 1$, FAHMM corresponds to SFA (Gopinath et al., 1998). Even though the state space distribution parameters are modelled explicitly, there are effectively

Table 1
Standard systems related to FAHMMs. FAHMM can be converted to the systems on the left hand side by applying the restrictions on the right.

System	Relation to FAHMMs
HMM	$M^{(x)} = 0$
SFA	$M^{(x)} = 1$
dynamic IFA	$M^{(o)} = 1$
STC	$k = p$ and $\boldsymbol{v}_t = \mathbf{0}$
Covariance EMLLT	$k > p$ and $\boldsymbol{v}_t = \mathbf{0}$

an equal number of free parameters in this FAHMM and SFA which assumes the state distribution with zero mean and identity covariance.

- By setting the number of observation space distribution components to one, $M^{(o)} = 1$, FAHMM corresponds to a dynamic version of IFA (Attias, 1999). The only difference to the standard IFA is the independent state vector element (factor) assumption which would require a multiple stream (factorial) HMM (Ghahramani and Jordan, 1997) with 1-dimensional streams in the state space. Effectively multiple streams can model a larger number of distributions but the independence assumption is relaxed in this FAHMM assuming uncorrelated factors instead of independent.
- By setting the observation noise to zero, $\boldsymbol{v}_t = \mathbf{0}$, and setting the state space dimensionality equal to the observation space dimensionality, $k = p$, FAHMM reduces to a semi-tied covariance matrix HMM. The only difference to the original STC model in (Gales, 1999) is that the mean vectors are also transformed in FAHMM.
- By setting the observation noise to zero, $\boldsymbol{v}_t = \mathbf{0}$, and setting the state space dimensionality greater than the observation space dimensionality, $k > p$, FAHMM becomes a covariance version of extended maximum likelihood linear transformation (EMLLT) (Olsen and Gopinath, 2002) scheme. FAHMM is based on a generative model which requires every state space covariance matrix being a valid covariance matrix; i.e. positive definite. EMLLT, on the other hand, directly models the inverse covariance matrices and allows "negative" variance elements as long as the resulting inverse covariance matrices are valid.

3 Implementation Issues

When factor analysed HMMs are applied for large vocabulary continuous speech recognition (LVCSR) there are a number of efficiency issues that must be addressed. As EM training is being used to iteratively find the ML estimates of the model parameters, an appropriate initialisation scheme is essential. Furthermore, in common with standard LVCSR systems, parameter tying may be used extensively. In addition, there is a large amount of matrix operations that need to be computed. Issues with numerical accuracy have to be considered. Finally, as there are two sets of hidden variables in FAHMMs, an efficient two level training scheme is presented.

3.1 Initialisation

One major issue with the EM algorithm is that there may be a number of local maxima. An appropriate initialisation scheme may improve the chances of finding a good solution. A sensible starting point is to use a standard HMM. A single Gaussian mixture component HMM can be converted to an equivalent FAHMM as follows

$$
\begin{aligned}
\boldsymbol{\mu}_j^{(x)} &= \boldsymbol{\mu}_{j[1:k]} \\
\boldsymbol{\Sigma}_j^{(x)} &= \frac{1}{2}\boldsymbol{\Sigma}_{j[1:k]} \\
\boldsymbol{C}_j &= \boldsymbol{I} \\
\boldsymbol{\mu}_j^{(o)} &= \begin{bmatrix} \mathbf{0} \\ \boldsymbol{\mu}_{j[k+1:p]} \end{bmatrix} \\
\boldsymbol{\Sigma}_j^{(o)} &= \begin{bmatrix} \frac{1}{2}\boldsymbol{\Sigma}_{j[1:k]} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_{j[k+1:p]} \end{bmatrix}
\end{aligned}
$$

where $\boldsymbol{\mu}_{j[1:k]}$ represent the first k elements of the mean vector and $\boldsymbol{\Sigma}_{j[1:k]}$ is the upper left k by k submatrix of the covariance matrix associated with state j of the initial HMM.

The above initialisation scheme guarantees that the average log-likelihood of the training data after the following iteration is equal to the one obtained using the original HMM. The equivalent FAHMM system can also be obtained by setting the observation matrices equal to zero and initialising the observation noise as the HMM output distributions. However, the proposed method can be used to give more weight on certain dimensions and should provide better convergence. Here it is assumed that the first k feature vector elements are the most significant. In the experiments, the state space dimensionality was chosen to be $k = 13$ which corresponds to the static parameters in a standard 39-dimensional feature vector. Alternative feature selection techniques such as Fisher ratio can also be used within this initialisation scheme.

As discussed in Section 2.1, the number of free parameters per FAHMM state, η, is the same as in a factor analysis model with Gaussian mixture distributions. Table 2 summarises the numbers of free parameters per HMM and FAHMM state discarding the mixture weights. The dimensionality of the state space, k, and the number of observation noise components, $M^{(o)}$, have the largest influence on the complexity of FAHMMs.

Table 2
Number of free parameters per HMM and FAHMM state, η, using $M^{(x)}$ state space components, $M^{(o)}$ observation noise components and no sharing of individual FAHMM parameters. Both diagonal covariance and full covariance matrix HMMs are shown.

System	Free Parameters (η)
diagonal covariance HMM	$2M^{(o)}p$
full covariance HMM	$M^{(o)}p(p+3)/2$
FAHMM	$2(M^{(x)}-1)k+pk+2M^{(o)}p$

When context-dependent HMM systems are trained the selection of the model set is often based on decision-tree clustering (Bahl, de Souza, Gopalkrishnan, Nahamoo, and Picheny, 1991). However, implementing decision-tree clustering for FAHMMs is not as straightforward as for HMMs. The clustering based on single mixture component HMM statistics is not optimal for HMMs (Nock, Gales, and Young, 1997). Since the FAHMMs can be viewed as full covariance matrix HMMs, decision-tree clustered single mixture component HMM models may be considered as a sufficiently good starting point for FAHMM initialisation. The initialisation of the context-dependent models can be done the same way as using standard context-independent HMMs described in Section 3.1.

In addition to state clustering, it is sometimes useful to share some of the individual FAHMM parameters. It is possible to tie any number of parameters between an arbitrary number of models at various levels of the model. For example, the observation matrix can be shared globally or between classes of discrete states as in semi-tied covariance HMMs (Gales, 1999). A global observation noise distribution could represent a stationary noise environment corrupting all the speech data. Implementing arbitrary tying schemes is closely related to those used with standard HMM systems (Young et al., 2000). The sufficient statistics required for the tied parameter are accumulated over the entire class sharing it before updating. If the mean vectors and the covariance matrices of the state space noise are tied on a different level, all the cross terms between the first-order accumulates and the updated mean vectors, $\hat{\boldsymbol{\mu}}_{jn}$, have to be used in the covariance matrix update formula. Equation (14), including all the cross terms, can be written as

$$\hat{\boldsymbol{\Sigma}}_{jn}^{(x)} = \mathrm{diag}\Bigg(\frac{\sum_{t=1}^{T} \sum_{m=1}^{M^{(o)}} \gamma_{jmn}(t) \Big(\hat{\boldsymbol{R}}_{jmn}(t) - \hat{\boldsymbol{x}}_{jmn}(t) \hat{\boldsymbol{\mu}}_{jn}^{(x)\prime} - \hat{\boldsymbol{\mu}}_{jn}^{(x)} \hat{\boldsymbol{x}}_{jmn}'(t) \Big)}{\sum_{t=1}^{T} \sum_{m=1}^{M^{(o)}} \gamma_{jmn}(t)} + \hat{\boldsymbol{\mu}}_{jn}^{(x)} \hat{\boldsymbol{\mu}}_{jn}^{(x)\prime} \Bigg)$$

where the first-order accumulates, $\sum_t \sum_m \gamma_{jmn}(t)\hat{\boldsymbol{x}}_{jmn}(t)$, may be different to those used for the mean vector update in Equation (13).

3.3 Numerical Accuracy

The matrix inversion described in Section 2.3 and the parameter estimation require many matrix computations. Numerical accuracy may become an issue due to the vast amount of sums of products. In the experiments it was found that double precision had to be used in all the intermediate operations to guarantee that the full covariance matrices were non-singular. Nevertheless, single precision was used to store the accumulates and model parameters due to the memory usage.

A large amount of training data is required to get reliable estimates for the covariance matrices in a LVCSR system. Sometimes the new variance elements may become too small for likelihood calculations. If any variance element becomes too small within the machine precision, a division by zero will occur. To avoid problems with FAHMMs the full covariance matrices in Equation (6) must be guaranteed to be non-singular. The matrix $\boldsymbol{C}_j \boldsymbol{\Sigma}_{jn}^{(x)} \boldsymbol{C}_j'$ is at most rank k provided the state space variances are valid. Therefore, it is essential that the observation noise variances are floored properly. In the experiments it was found that the flooring scheme usually implemented in HMM systems (Young et al., 2000) is sufficient for the observation variances in FAHMMs. With very large model sets the new estimates for the state space variances may become negative due to insufficient data for the component. In the experiments such variance elements were not updated.

3.4 Efficient Two Level Training

To increase the speed of training, a two level algorithm is adopted. The component specific first and second-order statistics form the sufficient statistics required in the parameter estimation described in Section 2.4. This can be verified by substituting the state vector statistics, $\hat{\boldsymbol{x}}_{jmn}(t)$ and $\hat{\boldsymbol{R}}_{jmn}(t)$, from

Equations (10) and (11) into the update Equations (12)-(15). The sufficient statistics can be written as

$$\tilde{\gamma}_{jmn} = \sum_{t=1}^{T} \gamma_{jmn}(t)$$
$$\tilde{\boldsymbol{\mu}}_{jmn} = \sum_{t=1}^{T} \gamma_{jmn}(t)\boldsymbol{o}_t$$
$$\tilde{\boldsymbol{R}}_{jmn} = \sum_{t=1}^{T} \gamma_{jmn}(t)\boldsymbol{o}_t\boldsymbol{o}_t'$$

Given these accumulates and the current model parameters, $\mathcal{M}$, the required accumulates for the new parameters can be estimated. Since the estimated state vector statistics depend on both the data accumulates and the current model parameters an extra level of iterations can be introduced. After updating the model parameters, new state vector distribution given the old data accumulates and the new model parameters can be estimated. These within iterations are guaranteed to increase the log-likelihood of the data. Fig. 2 illustrates the increase of the auxiliary function values during three full iterations, 10 within iterations each.

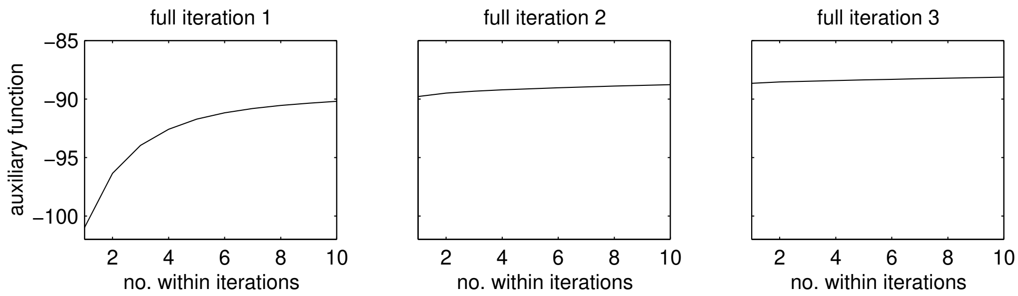

Fig. 2. Auxiliary function values versus within iterations during 3 full iterations of two level FAHMM training.

The efficient training algorithm can be summarised as follows

(1) Collect the data statistics using forward-backward algorithm;
(2) Estimate the state vector distribution $p(\boldsymbol{x}_t|j, m, n, \boldsymbol{O}, \mathcal{M})$;
(3) Estimate new model parameters $\hat{\mathcal{M}}$;
(4) If the auxiliary function value has not converged go to step 2 and update the parameters $\hat{\mathcal{M}} \rightarrow \mathcal{M}$;
(5) If the average log-likelihood of the data has not converged go to step 1 and update the parameters $\hat{\mathcal{M}} \rightarrow \mathcal{M}$.

The within iterations decrease the number of full iterations needed in training. The overall training time becomes shorter because less time has to be spent collecting the data accumulates. The average log-likelihoods of the training data against the number of full iterations are illustrated in Fig. 3. Four iterations of embedded training were first applied to the baseline HMM. The FAHMM system with $k = 13$ was initialised as described in Section 3.1. Both, one level training and more efficient two level training with 10 within iterations, were used and the corresponding log-likelihoods are shown in the figure.

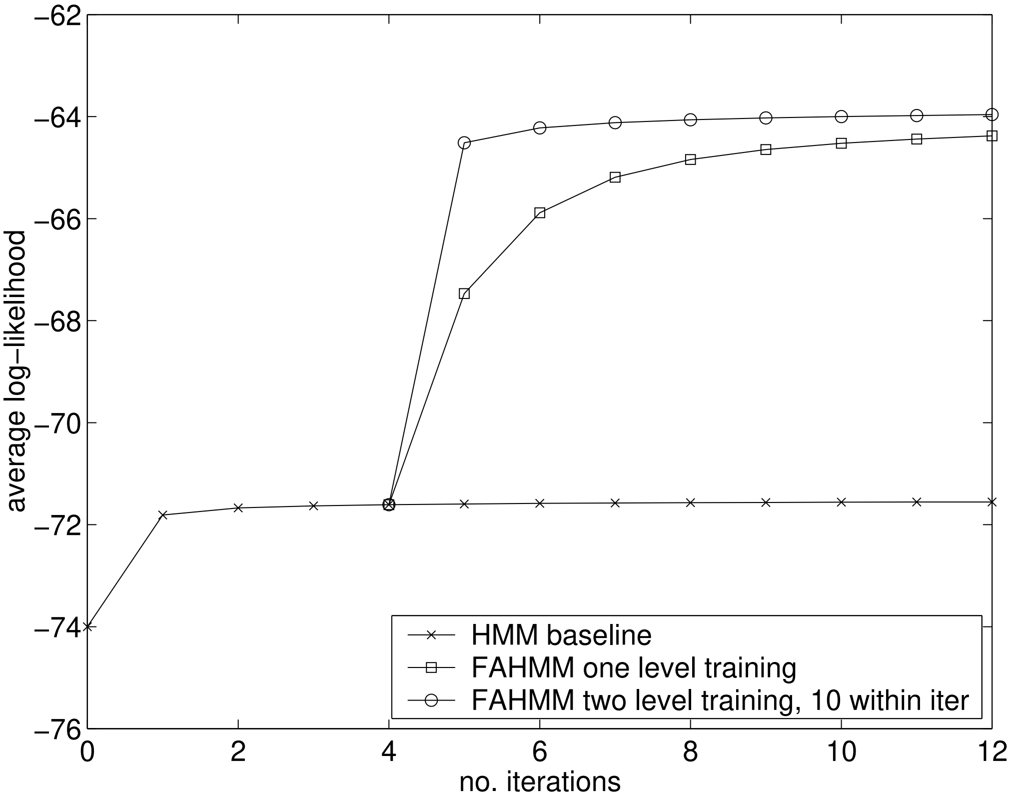

Fig. 3. Log-likelihood values against full iterations for baseline HMM and an untied FAHMM with $k = 13$. One level training and more efficient two level training with 10 within iterations were used.

4 Results

The results in this section are presented to illustrate the performance of some FAHMM configurations on medium to large speech recognition tasks. Only a small number of possible configurations have been examined and the configurations have not been chosen in accordance with any optimal criterion. Generally, configurations with fewer or equivalent number of free parameters compared to the baseline were chosen. The state space size, $k = 13$, was used since it was the number of static components in the chosen parameterisation. The aim was to show how FAHMMs perform with some possible configurations as well as compare them to standard semi-tied systems.

For initial experiments, a standard medium size speech recognition task, the ARPA Resource Management (RM) task, was used. Following the HTK "RM Recipe" (Young et al., 2000), the baseline system was trained starting from a flat start single Gaussian mixture component monophone system. A total of 3990 sentences {`train, dev_aug`} were used for training. After four iterations of embedded training, the monophone models were cloned to produce a single mixture component triphone system. Cross word triphone models that ignore the word boundaries in the context were used. These initial triphone models were trained with two iterations of embedded training after which a decision-tree clustering was applied to produce a tied state triphone system. This system was used as an initial model set for standard HMM, STC and FAHMM systems. A total of 1200 sentences {`feb89, oct89, feb91, sep92`} with a simple word-pair grammar were used for evaluation.

The baseline HMM system was produced by standard iterative mixture splitting (Young et al., 2000) using four iterations of embedded training per mixture configuration until no decrease in the word error rate was observed. The word error rates with the number of free parameters per HMM state up to 6 components are presented on the first row in Table 3, marked HMM. The best performance was 3.76% obtained with 10 mixture components. The number of free parameters per HMM state in the best baseline system was $\eta = 780$ per state. As an additional baseline a global semi-tied HMM system was built. The single mixture baseline HMM system was converted to the STC system by adding a global full 39 by 39 identity transformation matrix. The number of free parameters of the system increased by 1521 compared to the baseline HMM system. Since the number of physical states in the system was about 1600, the number of model parameters per state, η, increased by less than one. As discussed in Section 2.5, this STC system corresponds to a FAHMM with state space dimensionality $k = 39$ and observation noise equal to zero. The number of mixture components was increased by the mixture splitting procedure. Nine full iterations of embedded training were used with 20 within iterations and 20 row by row transform iterations (Gales, 1999). The results are presented on the second row in Table 3, marked STC. The best semi-tied performance was 3.83% obtained with 5 mixture components. As usual, the performance when using STC is better with fewer mixture components. However, increasing the number of mixture components in a standard HMM system can be seen to model the intra-frame correlation better.

A FAHMM system with state space dimensionality $k = 39$ and a global observation matrix, denoted as GFAHMM, was built for comparison with the STC system above. The global full 39 by 39 observation matrix was initialised to an identity matrix and the variance elements of the single mixture baseline HMM

Table 3
Word error rates (%) and number of free parameters, η, on the RM task, versus number of mixture components for the observation pdfs, for HMM, STC and GFAHMM systems.

System	$M^{(o)}$	1	2	3	4	5	6
HMM	η	78	156	234	312	390	**468**
	wer[%]	7.79	6.68	5.05	4.32	4.09	**3.99**
STC	η	78	156	234	312	**390**	468
	wer[%]	7.06	5.30	4.32	3.93	**3.83**	3.85
GFAHMM	η	117	195	273	351	**429**	507
	wer[%]	6.52	4.88	4.28	3.94	**3.68**	3.77

system were evenly distributed between the observation and state space variances as discussed in Section 3.1. The number of state space components was set to one, $M^{(x)} = 1$, and the observation space components were increased by the mixture splitting procedure. The system corresponds to a global full loading matrix SFA with non-identity state space covariance matrices. The number of additional free parameters per state was 39 due to the state space covariance matrices, which could not be subsumed into the global observation matrix, and 1521 globally due to the observation matrix. Nine full iterations of embedded training were used, each with 20 within iterations. The results are presented on the third row in Table 3, marked GFAHMM. The best performance, 3.68%, was achieved with 5 mixture components. The difference in the number of free parameters between the best baseline, $M^{(o)} = 10$, and the best GFAHMM system, $M^{(o)} = 5$, was 351 per state. Compared to the STC system, GFAHMM has only 39 additional free parameters per state. However, the GFAHMM system provides a relative word error rate reduction of 4% to the STC system.

These initial experiments show the relationship between FAHMMs and STC in practice. However, the training and recognition using full state space FAHMMs is far more complex than using global STC even though the observation matrix is shared globally. Since STC does not have observation noise, the global transform can be applied to the feature vectors in advance and full covariance matrices are not needed in the likelihood calculation. It should be noted that the errors the above three systems make are very similar. This was investigated by scoring the results of the systems against each other. The highest error rates for this cross evaluation were less than 2.50%. The performance of FAHMMs using lower dimensional state space is reported in the experiments below.

4.2 Minitrain

The Minitrain 1998 Hub5 HTK system (Hain, Woodland, Niesler, and Whittaker, 1999) was used as a larger speech recognition task. The baseline was a decision-tree clustered tied state triphone HMM system. Vocal tract length normalisation (VTLN) was used to make the system gender independent. Cross word triphone models with GMMs were used. The 18 hour Minitrain set defined by BBN (Miller and McDonough, 1997) containing 398 conversation sides of Switchboard-1 corpus was used as the acoustic training data. The test data set was the subset of the 1997 Hub5 evaluation set used in (Hain et al., 1999). The best performance, 51.0%, was achieved with 12 components which corresponds to $\eta = 936$ parameters per state. The mixture splitting was not continued further since the performance started degrading after 12 components.

A FAHMM system with state space dimensionality 13 was built starting from the single mixture component baseline system. An individual 39 by 13 observation matrix initialised as an identity matrix was attached to each state. The first 13 variance elements of the HMM models were evenly distributed among the observation and state space variances as discussed in Section 3.1. The mixture splitting was started from the single mixture component baseline system increasing the number of state space components while fixing the number of observation space components to $M^{(o)} = 1$. The number of observation space components of a single state space component system, $M^{(x)} = 1$, was then increased and fixed to $M^{(o)} = 2$. The number of the state space components was increased until no further gain was achieved and so on. The results up to the best performance per column are presented in Table 4. As discussed in Section 2.5, the row corresponding to $M^{(x)} = 1$ is related to a SFA system and the first column corresponding to $M^{(o)} = 1$ is related to a dynamic IFA without the independent state vector element assumption. The same performance as the best baseline HMM system was achieved using FAHMMs with 2 observation and 4 state space components, 51.0% ($\eta = 741$). The difference in the number of free parameters per state is considerable: the FAHMM system has 195 less than the HMM one. The best FAHMM performance, 50.7% ($\eta = 793$), was also achieved using fewer free parameters than the best baseline system, though the improvement is not statistically significant.

These experiments show how the FAHMM system performs in a large speech recognition task when a low dimensional state space was used. As the state space dimensionality and the initialisation were selected based on intuition, the results seem promising. Choosing the state space dimensionality automatically is very challenging problem, and it can be expected to improve the performance. Complexity control and more elaborate initialisation schemes will be studied in the future.

Table 4
Word error rates (%) and number of free parameters, η, on the Minitrain task, versus number of mixture components for the observation and state space pdfs, for FAHMM system with $k = 13$. The best baseline HMM word error rate was 51.0% with $M^{(o)} = 12$ ($\eta = 936$).

$M^{(x)}$	$M^{(o)}$	1	2	4
1	η	585	663	819
	wer[%]	53.3	51.7	51.0
2	η	611	689	845
	wer[%]	53.3	51.4	51.3
4	η	663	**741**	897
	wer[%]	53.0	**51.0**	50.9
6	η	715	**793**	949
	wer[%]	52.8	**50.7**	51.0
8	η	767	845	
	wer[%]	52.6	51.0	

4.3 Switchboard 68 Hours

For the experiments performed in this section, a 68 hour subset of the Switchboard (Hub5) acoustic training data set was used. 862 sides of the Switchboard-1 and 92 sides of the Call Home English were used. The set is described as "h5train00sub" in (Hain, Woodland, Evermann, and Povey, 2000). As with Minitrain, the baseline was a decision-tree clustered tied state triphone HMM system with VTLN, cross word models and GMMs. The 1998 Switchboard evaluation data set was used for testing. The baseline HMM system word error rates with the number of free parameters per state are presented on the first row in Table 5, marked HMM. The word error rate of the baseline system went down to 45.7% with 30 mixture components. However, the number of free parameters in such a system is huge, $\eta = 2340$ per state. The 14 component system was a reasonable compromise because the word error rate, 46.5%, seems to be a local stationary point. As an additional baseline a global semi-tied covariance HMM system was trained the same way as in the RM experiments. The results for the STC system are presented on the second row in Table 5, marked STC. The best performance, 45.7%, in the STC system was obtained using 16 components.

FAHMM system with state space dimensionality $k = 13$ was built starting from the single mixture component baseline system. The initialisation and mixture splitting were carried out the same way as in the Minitrain experiments in Section 4.2. Unfortunately, filling up a complete table was not feasible since the training time grows very long. The number of full covariance matrices defined in Section 2.3 is $M^{(o)}M^{(x)}$, and the memory is quickly filled when using effectively more than 16 full covariance matrices stored prior to training or recognition. Despite the efficient inversion and caching described in Section 2.3, the training and recognition times grow too long with the current implementation. The most interesting results here are achieved using only one state space component which corresponds to the SFA. The results are presented on the third row in Table 5, marked SFA. Increasing the number of state space components with a single observation space component, $M^{(o)} = 1$, (IFA) did not show much gains. This is probably due to the small increase in the number of model parameters in such a system. It is worth noting that the best baseline performance was achieved using FAHMMs with considerably fewer free parameters. The 12 component baseline performance, 46.7% ($\eta = 936$), was achieved by using FAHMMs with fewer parameters - namely $M^{(o)} = 2$ and $M^{(x)} = 8$ which corresponds to $\eta = 845$ free parameters per state.

Table 5
Word error rates (%) and number of free parameters, η, on the Hub5 68 hour task, versus number of mixture components for the observation pdfs, for HMM, STC, SFA and GSFA systems with $k = 13$. SFA is a FAHMM with a single state space mixture component, $M^{(x)} = 1$. SFA has state specific observation matrices whereas STC and GSFA have global ones.

System	$M^{(o)}$	1	2	4	6	8	10	12	14	16
HMM	η	78	156	312	468	624	780	936	**1092**	1248
	wer[%]	55.1	52.4	49.6	48.5	47.7	47.2	46.7	**46.5**	46.5
STC	η	78	156	312	468	624	780	936	1092	**1248**
	wer[%]	54.3	50.4	48.4	47.3	46.7	46.3	46.3	45.8	**45.7**
SFA	η	585	663	819	975	1131	1287	1443	**1599**	1755
	wer[%]	49.1	48.0	47.2	46.6	46.3	46.4	46.0	**45.8**	45.9
GSFA	η	91	169	325	481	637	793	949	1105	**1261**
	wer[%]	55.2	52.1	49.4	48.4	47.4	46.9	46.7	46.4	**46.1**

To see how the tying of parameters influence the results, a FAHMM system with state space dimensionality $k = 13$ and a global observation matrix $\boldsymbol{C}$ was built starting from the single mixture component baseline system as usual. The initialisation was carried out the same way as in the Minitrain experiments in Section 4.2. As before, filling up the table was not feasible due to the number of effective full covariance components in the system. Examining the preliminary results, the single state space component system appeared to be the most interesting. The results for the single state space component system are presented on the fourth row in Table 5, marked GSFA. The 12 observation component system achieved the same performance as the 12 component baseline system but further increasing the number of components proved to be quite interesting. The 16 observation space component system achieved 46.1% ($\eta = 1261$), the same performance as 24 component baseline system but with 611 free parameters fewer. It should also be noted that the STC system outperforms these configurations of FAHMMs in this task.

These experiments show that the current implementation of the FAHMM system has its limits when the task size is increased from the Minitrain task. The initialisation and choice of state space dimensionality require further attention, as previously indicated. The main contribution of these experiments was to show how an equivalent performance to HMMs can be achieved using dramatically fewer model parameters in a large speech recognition task with simple configurations of FAHMMs.

5 Conclusions

This paper has introduced the factor analysed HMM which is a general form of acoustic model. It combines a standard Gaussian mixture HMM with a shared and independent factor analysis model. FAHMM should provide a better model for the correlation between the feature vector elements compared to a standard diagonal covariance matrix HMM. It can be viewed as a compromise between diagonal and full covariance matrix systems. In addition, FAHMM can be viewed as a general state space model which allows a number of subspaces to be explored. A variety of configurations and sharing schemes, some of which correspond to standard systems, have been investigated. The ML estimation using EM algorithm is presented along with several schemes to improve both, time and memory efficiency. The speech recognition performance is evaluated in experiments using three medium to large vocabulary continuous speech recognition tasks. The results show that equivalent or slightly better performance compared to standard diagonal covariance Gaussian mixture HMMs can be achieved with considerably fewer model parameters. The FAHMM with 2 observation and 8 state space components gave performance equal to the best HMM systems for both Minitrain and Hub5 68 hour tasks.

Due to the flexibility of FAHMMs a large number of configurations can be explored. The number of Gaussian mixture components in both, the state and observation space, can be chosen. Different techniques to optimally choose the configuration have to be investigated. Another important question is how to choose an optimal state space dimensionality. These are standard problems in speech recognition and machine learning. The automatic complexity control for FAHMM based systems has to be addressed in the future.

Acknowledgements

A-V.I. Rosti is funded by an EPSRC studentship and Tampere Graduate School in Information Science and Engineering. He received additional support from the Finnish Cultural Foundation. This work made use of equipment kindly supplied by IBM under an SUR award.

References

Attias, H., 1999. Independent factor analysis. Neural Computation 11 (4), 803–851.

Bahl, L., de Souza, P., Gopalkrishnan, P., Nahamoo, D., Picheny, M., 1991. Context dependent modelling of phones in continuous speech using decision trees. In: Proceedings DARPA Speech and Natural Language Processing Workshop. pp. 264–270.

Chen, S., Gopalakrishnan, P., 1998. Clustering via the Bayesian information criterion with applications in speech recognition. In: Proceedings International Conference on Acoustics, Speech and Signal Processing. Vol. 2. pp. 645–648.

Gales, M., 1998. Maximum likelihood linear transformations for HMM-based speech recognition. Computer Speech and Language 12 (2), 75–98.

Gales, M., 1999. Semi-tied covariance matrices for hidden Markov models. IEEE Transactions on Speech and Audio Processing 7 (3), 272–281.

Gales, M., 2002. Maximum likelihood multiple subspace projections for hidden Markov models. IEEE Transactions on Speech and Audio Processing 10 (2), 37–47.

Ghahramani, Z., 1998. Learning dynamic Bayesian networks. In: Giles, C., Gori, M. (Eds.), Adaptive Processing of Sequences and Data Structures. Vol. 1387 of Lecture Notes in Computer Science. Springer, pp. 168–197.

Ghahramani, Z., Jordan, M., 1997. Factorial hidden Markov models. Machine Learning 29 (2-3), 245–273.

Gopinath, R., Ramabhadran, B., Dharanipragada, S., 1998. Factor analysis invariant to linear transformations of data. In: Proceedings International Conference on Speech and Language Processing. pp. 397–400.

Hain, T., Woodland, P., Evermann, G., Povey, D., 2000. The CU-HTK March 2000 HUB5E transcription system. In: Proceedings Speech Transcription Workshop.

Hain, T., Woodland, P., Niesler, T., Whittaker, E., 1999. The 1998 HTK system for transcription of conversational telephone speech. In: Proceedings

International Conference on Acoustics, Speech and Signal Processing. Vol. 1. pp. 57–60.

Harville, D., 1997. Matrix Algebra from a Statistician's Perspective. Springer.

Kumar, N., 1997. Investigation of silicon-auditory models and generalization of linear discriminant analysis for improved speech recognition. Ph.D. thesis, Johns Hopkins University.

Liu, X., Gales, M., Woodland, P., 2003. Automatic complexity control for HLDA systems. In: Proceedings International Conference on Acoustics, Speech and Signal Processing. Vol. 1. pp. 132–135.

Miller, D., McDonough, J., May 1997. BBN 1997 Acoustic Modelling, presented at Conversational Speech Recognition Workshop DARPA Hub-5E Evaluation.

Nock, H., Gales, M., Young, S., 1997. A comparative study of methods for phonetic decision-tree state clustering. In: Proceedings European Conference on Speech Communication and Technology. pp. 111–114.

Olsen, P., Gopinath, R., 2002. Modeling inverse covariance matrices by basis expansion. In: Proceedings International Conference on Acoustics, Speech, and Signal Processing. Vol. 1. pp. 945–948.

Rosti, A.-V., Gales, M., 2001. Generalised linear Gaussian models. Tech. Rep. CUED/F-INFENG/TR.420, Cambridge University Engineering Department, available via anonymous ftp from `ftp://svr-ftp.eng.cam.ac.uk/pub/reports/rosti_tr420.ps.gz`.

Roweis, S., Ghahramani, Z., 1999. A unifying review of linear Gaussian models. Neural Computation 11 (2), 305–345.

Saul, L., Rahim, M., 1999. Maximum likelihood and minimum classification error factor analysis for automatic speech recognition. IEEE Transactions on Speech and Audio Processing 8 (2), 115–125.

Viterbi, A., 1967. Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. IEEE Transactions on Information Theory IT-13, 260–269.

Young, S., Kershaw, D., Odell, J., Ollason, D., Valtchev, V., Woodland, P., 2000. The HTK Book (for HTK Version 3.0). Cambridge University.