CAMBRIDGE UNIVERSITY

ENGINEERING DEPARTMENT

LOOSELY COUPLED HIDDEN MARKOV MODELS FOR AUTOMATIC SPEECH RECOGNITION IN NOISY ACOUSTIC CONDITIONS

CUED/F-INFENG/TR.449

Thaxila Karunatillake and Steve Young

February 2003

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

E-mail: jtw28@eng.cam.ac.uk
http://www-svr.eng.cam.ac.uk/~jtw28

Abstract

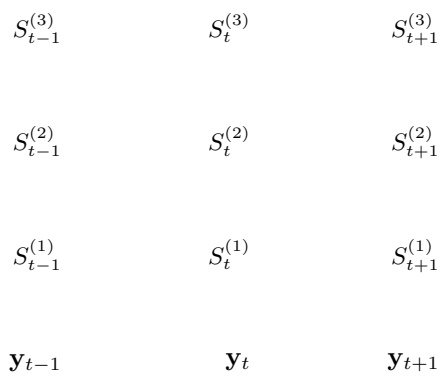
This work presents improvement in speech clustering, segmentation, and speech recognition that can be achieved in noisy acoustic conditions by modelling the environment and speech separately using Loosely Coupled Hidden Markov Models. It presents evidence that the commonly used HMM model is an inadequate model for speech recognition in noisy acoustic conditions. Several FHMM models that increase in complexity are studied and applied to speech clustering, segmentation and speech recognition with increased accuracy.

1 Introduction

In most current state of the art speech recognition systems the speech stream is modelled by a serial combination of the proverbial first order Hidden Markov Model. This model has proven to work well in clean speech conditions where planned speech exists in a clear acoustic environment. One such task is speech recognition of Wall Street Journal.

However, speech differs from these ideal conditions in two ways which leads to performance degradation in recognition. Firstly, degradation is observed where speech is conversational and not planned. Limited performance improvements have been observed by modelling conversational speech by coupled higher order HMM's [1] [2]. Secondly performance degradation occurs due to the existence of noise as background to speech. This thesis examines the improvements that can be achieved by modelling the noise sources and speech by higher order HMM's. The techniques are investigated for training large vocabulary continuous speech recognition systems for speech in noisy acoustic environments, and applied to the task of speech recognition in Broadcast News.

Consider, for instance, the application of Broadcast News. [65] The transcription provides information on 3 background sources - background speaker, background music and other background noise. This information can be modelled by a 3^{rd} order coupled Factorial Hidden Markov Model (FHMM) as in the figure below, in which the source streams correspond to models of the background speaker, background music, and foreground speech respectively. Any other noise is grouped together and modelled by a 'mixing noise' model. Even more complexity may exist, where loud noise may influence the stress level of the speaker and hence provide coupling with the speech model.



Learning the higher order HMM consists of two stages: inference and parameter learning. The learning process is carried out using the Expectation Maximization (EM) algorithm, where the Expectation (E) step corresponds to solving the Inference problem, and the Maximization (M) step corresponds to the parameter Learning problem. The above 2 stage process is carried out assuming that the structure of the model is given. The problem of learning the optimal structure for the given problem, structure learning, is itself an important part of parameter learning. In the above example the structure is inferred by using prior knowledge provided with the data.

The EM algorithm iterates between the E-step which fixes the current parameters and computes the posterior probabilities of the latent variable (hidden) states $P(\{S_t\}|\{\mathbf{y}_t\})$, and the M-step that fixes the current latent state occupation probabilities and re-learns the parameters, both in order to maximize the expected log-likelihood of the observations. The exact M-step for the FHMM is simple and tractable. However, the resulting state space explosion of the FHMM renders the exact computation of the E-step computationally intractable. Hence it is necessary to employ approximations for the interpolation of higher order FHMM's for the E-step. This thesis researches a number of techniques subject to current research.

- Variational Approximations (Gaharamani & Jordan) [4]
- Stochastic Sampling Approximations: Markov Chain Monte Carlo methods (Kanazawa, Koller & Russell [47], Radford Neal [48])
- Variational, Stochastic Sampling and Exact technique combination [45]
- Learning with Boltzmann Chains (William & Hinton, Saul & Jordan, MacKay)
- Mixed Memory HMM Approximation (Saul & Jordan)
- Dynamic HMM Model Selection (Hain & Woodland) [56]

The main research concentrates on

i. Efficient approximation techniques for solving the inference problem (E-step) for learning the FHMM.

ii. Regularization of the EM algorithm for better generalization in learning the FHMM. Structure learning and model selection for learning the optimal structure for the FHMM for the given problem.

iii. Improvement in ASR achievable by the use of the FHMM as a predictive to increasingly generative model of noise and speech sources.

iv. Improved speech clustering, segmentation and meta-data markup using FHMM models for frame classification and segmentation

Noise Robustness in Speech Recognition [50] [51] [52] [53] [54] [55]

There are three levels of degradation in spontaneous speech recognition that occur with the existence of background noise in the acoustic environment. The baseline performance for comparison is taken with respect to the conditions where both training and test data of spontaneous speech for the speech recognition system is recorded in a laboratory environment free of any background noise.

The first level of performance degradation occurs when the available training data and test data both occur in an environment with noise, where the noise characteristics and the noise levels are the same (matched) for both data sets. The techniques outlined in the thesis are Model Based techniques that seek to bridge the performance gap between speech recognition in such matched noisy conditions and baseline speech recognition performances in clear laboratory conditions.

e.g. There is a performance gap between spontaneous speech recognition of speech in a quiet recording studio by a system trained on similar data, and spontaneous speech recognition in an office environment trained with speech in the same office environment.

Speech recognition performance degrades further when there is a mismatch between the noise conditions in the training data and the test data. The mismatch may be twofold:

i. Mismatch in Noise Type: e.g. the training data is recorded in an office environment while the test data is recorded in a moving vehicle.

ii. Mismatch in Noise Level or SNR (Speech to Noise Ratio): e.g. the training data is recorded with a close microphone in a vehicle while the test data is recorded with a microphone further away from the speaker in the same vehicle.

There are several general techniques currently used for achieving Noise Robustness in speech recognition work.

1. Noise Robust Feature Extraction:
   These techniques seek to filter out the noise signals from the foreground speech signal at the feature extraction level. Signal filtering and noise reduction techniques are commonly employed during or prior to feature extraction. There are front-end filtering techniques applied to the initial speech signal prior to feature extraction (e.g. Weiner filtering), during feature extraction (cepstral mean normalization, variance normalization), and post feature extraction (e.g. ARMA filters). Ideally, if the clean speech signal can be fully extracted, these techniques could address all types of speech recognition degradations outlined above.

2. Microphone Array Processing:
   Blind Source Separation techniques (with multiple sensors) are employed to separate the speech signal from the background noise signals. For these techniques both training and test speech is recorded with several microphones, the number of microphones being roughly equivalent to the number of expected main speech and noise signal sources. The separated speech signal is then used for training and recognition.

3. Model Adaptation to the Environment:
   Both of the above techniques process the speech signal prior to training and testing. The model based techniques seek to adapt the models to the environment in order to improve recognition rates by modelling both speech and noise conditions in the new noise environment. Model adaptation techniques vary according to the two kinds of adaptation data available:

   (a) Model adaptation using Noisy Speech data as adaptation data. The models are adapted to the noise condition using techniques such as MAP and MLLR.

   (b) Model adaptation using only Noise data as adaptation data. A separate Noise model is trained on the noise data alone and composed with a clean speech model. Multiple models corresponding to multiple SNR rates can be composed to suit unknown test SNR's. Parallel Model Combination is one such model composition technique.

Assume that there is adequate, single microphone recorded, training and test data in a particular noise environment. (e.g. The Broadcast News Corpus) Both test and training data are matched in noise type and SNR. Hence, there can be no further improvements achievable by the model environmental adaptation techniques.

The hypothesis subject to experimentation is that the first order HMM is an inadequate model of speech and background noise in this situation, and that speech recognition performances in matched conditions can be further improved by the higher order FHMM's presented in this thesis, that better represent the multiple sources generating the signal.

In the first instance evidence that the first order HMM is an inadequate model for representation of speech in noise is presented within a framework of speech segment clustering assessment with respect to speech and the environment, and acoustic environment identification.

FHMM Models that explicitly represent the environment and speech are then explored, with the complexity of the models increasing from a simple predictive model to a highly complex generative model. The increase in the complexity of the models is threefold:

1. The number of source streams increases from a single stream in an HMM model to multiple streams.

2. The source distribution(s) will increase from a discrete distribution to a single Gaussian distribution and in turn, to a Mixture of Gaussian distribution that is capable of representing any distribution to arbitrary accuracy. Hence the distributions become more accurate models of signal/noise sources.

3. The dynamics generating the dependency between the sources and the observation model will be shown to be non-linear in the cepstral domain. The modelling of this dependency will increase from a simple Bayesian dependency, to a linear dependency, to a true non-linear dependency with an added stochastic noise model.

It is my expectation that the model's performance and predictive ability will increase up to a point as the complexity of the model increases, but will level off or fall thereafter. This is due to the principle known as Occam's Razor, that the simplest possible explanation is the best. It is a search for a model of speech in noisy acoustic conditions that is as simple as possible, but no simpler.

The performance improvements achievable by explicitly modelling noise using the proposed FHMM models compete against the improvements achievable by noise robust front-end processing techniques. These noise robust front-end processing techniques have two problems. Firstly noise may be additive and convolutional; these two types of noise corrupt the speech signal in very different ways, and it may be difficult to identify and filter both noise corruptions with filtering. Secondly, distortion by noise over estimation and residual corruption by noise under-estimation cause spectral distortion and degrades speech recognition performance. These problems may be avoided by an explicit noise model. The time and resource costs of the training and recognition algorithms for FHMM modelling should be compared against the costs of using successful front-end feature extraction techniques.

Model Notation

T	the length of the time series
M	the number of streams in the FHMM model
$K^{(m)}$	the number of hidden states modelling stream m
D	the dimensionality of the observations or the number of sensor signals
$S_t^{(m)}$	state occupied by stream (m) at time t: a vector of length $K^{(m)}$ with 1 indicating the occupied state and 0 elsewhere
S_t	the set of M states occupied by each one of the streams at time t
$\{S_t\}$	the set of TM states occupied by each stream for the T time instances
$x_t^{(m)}$	the real valued source signal of stream (m) at time t
$\mathbf{x}_t$	the M length vector of real valued source signals at time t
$y_t^{(d)}$	the real valued signal of the d^{th} sensor
$\mathbf{y}_t$	the D length vector of real valued observation (or sensor signals) at time t
$\{\mathbf{y}_t\}$	the T length set of all observations
$\{\mathbf{y}_t, S_t\}$	the $T + TM$ length set of all observations and hidden states
$\langle S_t^{(m)} \rangle$	the $K^{(m)}$ vector of state occupation probabilities of stream m at time t
$\langle S_{t-1}^{(m)} S_t^{(m)'} \rangle$	the $K^{(m)} \times K^{(m)}$ matrix of state transition probabilities in stream m from time t to $t+1$
$\langle S_t^{(m)} S_t^{(n)'} \rangle$	the $K^{(m)} \times K^{(m)}$ matrix of state transition probabilities between 2 streams at time t

2 The Structure and Setting of the Problem

The FHMM model is shown in Figure 1.

- The model is a time series model, for time instances $t = 1 \dots T$.
- The observation at time t is a D-dimensional vector $\mathbf{y}_t$.
- There are $m = 1 \dots M$ sources. The sources at each time instance t are represented by M hidden state variables
  $S_t = (S_t^{(1)}, \dots, S_t^{(m)}, \dots, S_t^{(M)})$.
- The hidden states observe the Markov property such that each state variable is independent of the other state variables, given the preceding state variable.
  $P(S_t|S_{t-1}) = \prod_{m=1}^{M} P(S_t^{(m)}|S_{t-1}^{(m)})$
- The coupling between the observation and the m^{th} source stream is carried out according to some mixing matrix/matrices $W^{(m)}$.
- The hidden state variables at time t, though marginally independent, becomes conditionally dependent given the observation at time t. Hence, the dependency with the observation effectively couples all of the hidden state variables.

2.1 The Expectation Maximization Algorithm

The algorithm employed for learning the FHMM is the Expectation Maximization (EM) algorithm [42]. The EM algorithm consists of a two step procedure, iterated until convergence.

The E-step holds the parameters of the current model constant and computes the complete dataset for the model. Computing the expected state occupation (posterior)probabilities of the complete dataset is defined as the Inference Problem. The Inference Problem for the FHMM is intractable for models having more than 2 sources. Hence approximation techniques must be employed for this step.

The M-step consists of re-learning the parameters of the model using the complete dataset, in a way which maximizes the log likelihood of the observed data. Given the complete dataset, this parameter learning problem for the FHMM is simple and tractable.

In addition to the parameters of the FHMM, and additional noise model is necessary to represent the additional noise sources not explicitly modelled by the FHMM. This is mixing noise, and must be estimated as part of the EM process.

The problem takes on varied, but related, characteristics depending on the nature of the sources.

2.2 The FHMM modelling variation in the source/sensor means with discrete distributions

[4] The sources are modelled as discrete random variables $S_t^{(m)}$ that can assume one of $K^{(m)}$ discrete values. The mean of the single Gaussian which models the conditional probability of the observation $P(\mathbf{y}_t|S_t)$ is given by:

$$\mu_t = \sum_{m=1}^{M} W^{(m)} S_t^{(m)}$$

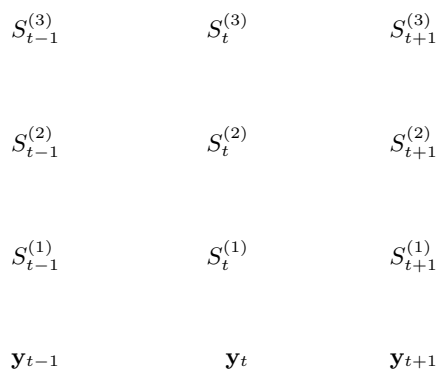


Figure 1: The Factorial Hidden Markov Model

In this model, each $S_t^{(m)}$ is a $K^{(m)}$-dimensional indicator vector that selects one of the k columns of the mixing matrix $W^{(m)}$. The sum of all M selected D-dimensional vectors forms the mean of the observation, conditional on the sources.

The resulting probability density for a D dimensional observation $\mathbf{y}_t$, given the sources, is

$$P(\mathbf{y}_t|S_t) = \frac{1}{\sqrt{(2\pi)^D|\Sigma|}}\ exp\left\{-\frac{1}{2}(\mathbf{y}_t - \mu_t)'\Sigma^{-1}(\mathbf{y}_t - \mu_t)\right\}$$

There are $K^{(m)}$ setting for each of the M state variables, giving approximately K^M possible mean vectors. Taking the sum over hidden state variables, the resulting marginal density gives an observation $\mathbf{y}_t$ that is a Mixture of Gaussian's with K^M mixture components and a constant covariance Σ.

This is a predictive model of the acoustic signal, rather than a generative model, as the sources do not correspond to accurate models of acoustic signal sources in real life. Acoustic signals can not be effectively modelled by discrete random variables. The interest in this model is to investigate whether predictive improvements in LVCSR result from the high representational power of the network.

2.3 The FHMM modelling continuous Gaussian source/sensor signals

Each source $S^{(m)}$ is modelled by a single Gaussian random variable with mean $\mu^{(m)}$ and covariance $\Sigma^{(m)}$. The sources are now mixed by a single $D \times M$ mixing matrix $\mathbf{W}$. The sources and conditional probability $P(\mathbf{y}_t|S_t)$ is now given by:

$$\begin{aligned} p(x_t^{(m)}) &= \mathcal{N}(x_t^{(m)} : \mu^{(m)}, \nu^{(m)}) \\ y_t &= \mathbf{W}x_t + u_t \end{aligned}$$

This inference problem corresponds to the **Blind Source Separation** problem using **Factor Analysis** (FA) [26] or **Probabilistic Principal Component Analysis** (PPCA) [27].

In the solution using PPCA the mixing is assumed to be noiseless or the mixing noise covariance must be of the form $\sigma^2\mathbf{I}$; e.g. $n_t \sim \mathcal{N}(0, \sigma^2\mathbf{I})$. When the mixing noise is uncorrelated, having diagonal covariance but which is not necessarily of the form $\sigma^2\mathbf{I}$, FA is employed for the solution.

In both cases FA and PPCA use only second order statistics of the observed data to perform estimations. This requires the factors or the sources to be merely uncorrelated, but not necessarily independent. As a result the noise source and factor mixing matrices can not be identified uniquely, but to an arbitrary rotation only. [19]

2.4 The FHMM modelling continuous Mixture of Gaussian source/sensor signals

Each source is modelled by a mixture of Gaussian's with $C^{(m)}$ components. The mean single Gaussian conditional observation density $P(\mathbf{y}_t|S_t)$ is given by:

$$\begin{aligned} p(x_t^{(m)}) &= \sum_{c=1}^{C^{(m)}} \gamma^{(mc)} \mathcal{N}(x_t^{(mc)} : \mu^{(mc)}, \nu^{(mc)}) \\ y_t &= \mathbf{W} x_t + u_t \end{aligned}$$

This inference problem also corresponds to the Blind Source Separation problem using **Independent Component Analysis** (ICA) [24] or **Independent Factor Analysis** (IFA) [19]. However both ICA and IFA, defined for instantaneous mixing, must be extended to solve the dynamic time-series problem at hand.

ICA implements a restricted version of the problem that allows only square, invertible mixing and no noise. IFA provides a solutions that allows noise of any form and non-square mixing. IFA also provides a solution that uniquely determines the mixing matrices and the factors, giving a unique solution to the BSS problem.[19]

The model becomes an increasingly generative model, as the sources may now be uniquely determined and may accurately represent acoustic and noise signals in real life.

A theoretical extension to the problem of dynamic time-series modelling and a comparative study of performance improvements in LVCSR against the costs of the algorithms for each model is presented.

2.5 Regularization and Model Selection

Structure Learning: A problem that occurs with these models is the selection of the optimal number of sources. In the discrete case the optimal number of discrete values K achievable by the discrete random variables must also be selected. The cost of selecting these parameters experimentally, using a method such as cross-validation, is prohibitively expensive. Automatic model selection methods and algorithms are investigated to address this problem. Alternatively, prior knowledge about the acoustic data can be used for effective structure learning.

Regularization: Secondly, the algorithm for model training is EM, which always favors models with a large number of parameters. This renders a model liable to overfit the training data. Specially in the case of the FHMM which witnesses a large increase in the number of parameters with each additional source in the model, regularization of the EM algorithm is very important. Techniques of regularization for the EM algorithm, such as penalizing for increasing the number of parameters, are investigated to solve this problem.

A number of model selection and regularization criteria such as BIC, AIC, CAIC and MDL will be subject to experimentation.[43]

Novel algorithms and techniques as well as improvements to the current algorithms and regularization techniques are investigated after an initial comparative study.

2.6 Variational Approximation Techniques

[9] Variational Approximation techniques address the problem of a complex graphical model with a distribution that cannot be tractably inferred, by approximating the model with a graphical model that is simpler and has a distribution that can be tractably inferred. For a given FHMM, a simpler graphical model and a new tractable distribution $Q(\{S_t\})$ over the hidden states which

Name	Author	Year	Penalty
AIC	Akaike	1972	k
BIC	Schwarz	1978	$\frac{k}{2}\log n$
CAIC	Bozdogan	1987	$\frac{k}{2}\log n + \frac{k}{2}$
CAICF	Bozdogan	1987	$k + \frac{k}{2}\log n + \frac{1}{2}\log \mid I(\theta) \mid$
MDL	Rissanen	1987	$\frac{k}{2}\log n + (\frac{k}{2}+1)\log(k+2)$
MML	Wallace and Freeman	1987	$\frac{d}{2}(1+\log \kappa_d) + \frac{1}{2}\log \mid I(\theta) \mid$
Notation:			
k	Number of free parameters in the model.		
n	Size of the data sample.		
d	Dimension of the data space.		
$I(\theta)$	Fisher Information matrix of the model.		
κ_d	Constant of the optimal d-dimensional quantizing lattice.		

Table 1: Model Dimension Estimates.

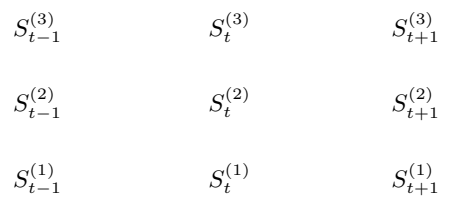


Figure 2: The Mean Field Approximation

is a lower bound on the log likelihoods of the observations, is defined. $Q(\{S_t\})$ approximates the intractable posterior distribution $P(\{S_t\}|\{\mathbf{y}_t\})$ and the parameters are learned in order to minimize the divergence between $Q(\{S_t\})$ and the exact posterior. It can be shown that this process maximizes the log likelihoods of the observed data. It can also be shown that any distribution $Q(\{S_t\})$ is a lower bound on the log likelihood of the observations.

$$KL(Q||P) = \sum_{\{S_t\}} Q(\{S_t\}) log\left[\frac{Q(\{S_t\})}{P(\{S_t\}|\{\mathbf{y}_t\})}\right]$$

As a result of this process the EM algorithm that uses a variational approximation for the E-step assures that a lower bound on the true likelihood is being maximized.

2.6.1 Mean Field Approximation

[4] The Mean Field Approximation approximates that all state variables are independent.

The distribution over the hidden states is defined as:

$$Q(\{S_t\}|\theta) = \prod_{t=1}^{T}\prod_{m=1}^{M} P(S_t^{(m)}|\theta_t^{(m)})$$

The variational parameters $\theta = \{\theta_t^{(m)}\}$ are the means of the states; hence the name of the approximation.

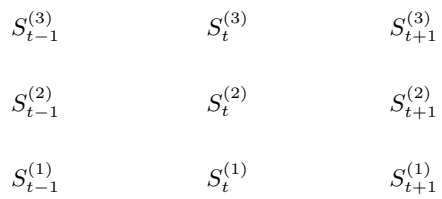


Figure 3: The Structured Variational Approximation

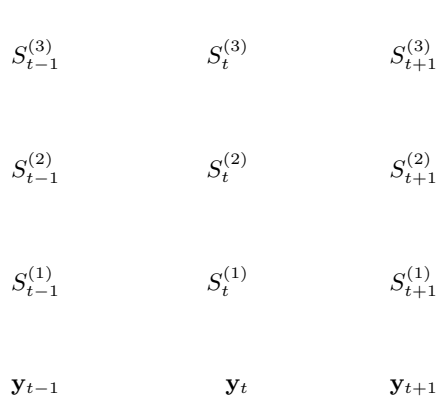


Figure 4: The Forest of Trees Approximation

2.6.2 Structured Variational Approximation

[4] The Structured Variational Approximation assumes that the state variables retain the Markov property within each chain, but are independent across the chains.

The distribution over the hidden states is defined as:

$$Q(\{S_t\}|\theta) = \frac{1}{Z_Q} \prod_{m=1}^{M} P(S_1^{(m)}|\theta) \prod_{m=1}^{M} P(S_t^{(m)}|S_{t-1}^{(m)}, \theta)$$

where Z_Q is the normalization constant that ensures Q integrates to one.

2.6.3 Forest of Trees Approximation

[8] The distribution over the hidden states is defined as:

$$Q(\{S_t\}|\theta) = \frac{1}{Z_Q} \prod_{t=1}^{T} P(\mathbf{y}_t|S_t) \prod_{m=1}^{M} P(S_t^{(m)})$$

where Z_Q is the normalization constant that ensures Q integrates to one.

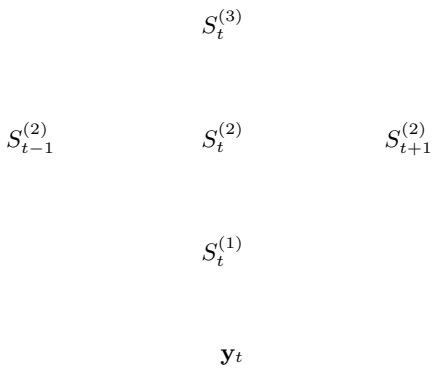


Figure 5: The Markov Blanket of FHMM node $S_t^{(2)}$

2.7 Sampling Techniques: Markov Chain Monte Carlo Estimations

[48] The FHMM is a highly expressive graphical model. However, the probability distributions complex GM's give rise to, such as the posterior distribution over the hidden variables of a FHMM, are very complex with probabilities varying greatly over a very high dimensional space. The complexity makes the exact inference of such distributions intractable. Often, however, a sample of points drawn from such a distribution can give a satisfactory picture of it. As the number of samples increases, the approximating distribution converges to the true distribution. MCMC techniques approximate a highly complex distribution by updating a an initial 'guessed' distribution according to samples drawn from the required distribution until convergence. Hence, MCMC techniques can be used to approximate statistics associated with the joint probability distribution $P(\{S_t, \mathbf{y}_t\})$.

It is difficult to sample from the complex distribution $P(\{S_t, \mathbf{y}_t\})$. Exploring the independence relations it can be shown that a given node is conditionally independent of all other nodes given its 'Markov Blanket', which is given by the node's parent, children and co-parent nodes. The Markov Blanket for a hidden node in the FHMM is shown in Figure 5. Hence, sampling from each node can be done with ease since $S_t^{(m)}$ sampled from the Markov Blanket $P(S_t^{(m)}|S_t^{(n)} : n \neq m, S_{t-1}^{(m)}, S_{t+1}^{(m)}, \mathbf{y}_t)$ is equivalent to $P(S_t^{(m)}|S_{t-1}^{(m)})\ P(S_{t+1}^{(m)}|S_t^{(m)})\ P(\mathbf{y}_t|S_t^{(1)} \dots S_t^{(M)})$.

The sampling process using the MCMC technique Gibb's Sampling is as follows. [8] [48]

The set of variables concerned is $S = \{S_1^{(1)}, S_1^{(2)}, \dots, S_t^{(m)}, \dots, S_T^{(M)}, \mathbf{y}_1, \dots, \mathbf{y}_T\}$. The variables in the set are addressed as S_i. It is necessary to calculate the statistics associated with the joint probability distribution $P(S)$:

- Marginal distributions such as $P(S_t^{(m)}), P(S_{t-1}^{(m)}, S_t^{(m)})$
- Conditional probabilities such as the posteriors $P(S_t^{(m)}|\mathbf{y}_t)$ and $P(S_t^{(m)}|S_{t-1})$
- Likelihoods $P(\mathbf{y}_t)$

Sampling methods generate samples from $P(S)$ and compute empirical statistics. However, it is difficult to sample from $P(S)$. The solution is to set up a simple Markov Chain simulation with equilibrium distribution $P(S)$. The Markov Chain:

- Initializes S_i to arbitrary values
- Chooses i randomly
- Samples from $P(S_i|S \setminus S_i)$, by sampling from the Markov Blanket of S_i which is much smaller than $P(S \setminus S_i)$, and updates statistics.
- Iterates sampling and update

Sampling once from each of the $T \times M$ hidden variables in the model results in a new sampling

of the hidden states of the model. The sequence of overall states resulting from each sampling pass defines a Markov chain over the state space of the model. This Markov chain is guaranteed to converge to the posterior probabilities of states, given the observations. After some suitable time, samples from the Markov chain can be taken as approximate samples from the posterior probabilities. The first and second order statistics required for the E-step are collected during this sampling process.

Difficulties:

[48] [4] The time to convergence can be very slow. Specially in the case of LVCSR where the number of states can be very high, this can become a real problem. Sampling is used in the E-step of the EM algorithm where there can be a time tradeoff between the number of samples used and the number of EM iterations. It is wasteful to wait until full MCMC convergence early on in learning (in the early E-steps), when the distribution from which the samples are drawn from is far from the posterior given the optimal parameters. In practice it has been found that approximate early E-steps using a few Gibbs samples (e.g. about ten samples for each hidden variable) tends to increase the likelihood. Only the latter E-steps need achieve true MCMC convergence.

More efficient and sophisticated sampling processes such as 'slice sampling', that allow the statistics update of multiple variables simultaneously and adapt automatically to the characteristics of the distribution, can be explored for increased efficiency.

Convergence can be hard to diagnose, and it becomes necessary to use reasonable heuristics for this.

2.7.1 MCMC Techniques

1. Gibbs Sampling
2. The Metropolis Algorithm
3. The Hybrid Mote Carlo Algorithm
4. Simulated Annealing, Free Energy Estimation and Parallel Implementations
   These are refinements to ensure faster convergence and to determine convergence

2.8 Combining Exact, Variational and MCMC Techniques

[45] An alternative to using MCMC techniques with incomplete convergence to estimate a complex distribution, is to use MCMC to approximate a variationally approximate distribution during the initial EM iterations. These variational distributions can be learned exactly, rather than using MCMC, for the initial iterations. The exact posterior distribution is then used as the target distribution during the final few EM iterations for which achieving near MCMC convergence is attempted.

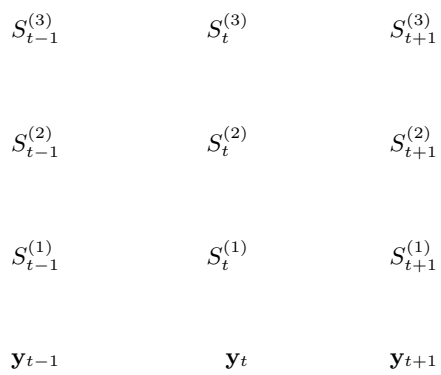


Figure 6: The Factorial Hidden Markov Model

3 Learning the FHMM Modelling Source and Sensor Signal Means

In this predictive model of speech the latent variables $S_t^{(m)}$ are discrete random variables that can take on one of K values. (Selection of K is part of the model selection problem.)

3.1 The FHMM Probability Model for Source/Sensor Mean Modelling

[4] The properties of the FHMM, shown in figure 1, are summarized as follows for the case where the latent variables are discrete random variables :

- The models is a time series model, for time instances $t = 1 \dots T$.
- The conditional distribution for the observation, given the latent variables, at time t is a D-dimensional vector $\mathbf{y}_t$ modelled by the Gaussian:

$$\begin{aligned} P(\mathbf{y}_t|S_t) &= \mathcal{N}(\mathbf{y}_t : \mu_t, \Sigma) \\ &= \frac{1}{\sqrt{(2\pi)^D|\Sigma|}}\, exp - \frac{1}{2}\left\{(\mathbf{y}_t - \mu_t)'\,\Sigma^{-1}\,(\mathbf{y}_t - \mu_t)\right\} \end{aligned} \tag{1}$$

- There are $m = 1 \dots M$ sources. The sources at each time instance t are represented by M hidden state variables that can each assume one of K values
  $S_t = (S_t^{(1)}, \dots, S_t^{(m)}, \dots, S_t^{(M)})$.
- The hidden states observe the Markov property such that each state variable is independent of the other state variables, given the preceding state variable.

$$P(S_t|S_{t-1}) = \prod_{m=1}^{M} P(S_t^{(m)}|S_{t-1}^{(m)}) \tag{2}$$

- Using the independence relations, the joint probability for the sequence of states and observations can be factorized as:

$$P(\{S_t, \mathbf{y}_t\}) = P(S_1)\; P(\mathbf{y}_1|S_1)\; \prod_{t=2}^{T} P(S_t|S_{t-1})\; P(\mathbf{y}_t|S_t) \tag{3}$$

- The coupling between the observation and the m^{th} source stream is carried out according to $D \times K$ mixing matrix $W^{(m)}$.

$$\mu_t = \sum_{m=1}^{M} W^{(m)} S_t^{(m)} \tag{4}$$

- The hidden state variables at time t, though marginally independent, become conditionally dependent given the observation at time t. Hence, the dependency with the observation effectively couples all of the hidden state variables.

- The marginal distribution for the observation is obtained from (1) by summing over all possible states. There are K settings for each of the M state variables; hence there are K^M possible mean vectors obtained by forming sums of M columns chosen from each of the $W^{(m)}$ matrices. The resulting marginal distribution is a Gaussian mixture model:

$$P(\mathbf{y}_t|\theta) = \sum_{c=1}^{K^M} p_c \, \mathcal{N}(\mathbf{y}_t : \mu_{tc}, \Sigma)$$
$$P(\{\mathbf{y}_t\}|\theta) = \prod_{t=1}^{T} \sum_{c=1}^{K^M} p_c \, \mathcal{N}(\mathbf{y}_t : \mu_{tc}, \Sigma) \tag{5}$$

- The parameters of the model are

$$\begin{aligned} \theta &= \{W^{(m)}, \pi^{(m)}, \xi^{(m)}, \Sigma\} \qquad (6) \\ \text{where,} \quad \pi^{(m)} &= P(S_1^{(m)}) \\ \xi^{(m)} &= P(S_t^{(m)}|S_{t-1}^{(m)}) \end{aligned}$$

- The complete dataset for FHMM learning therefore includes the observations $\{\mathbf{y}_t\}$ and the following expected likelihoods (for all t and m) given the current model parameters θ :

$$\begin{aligned} \langle S_t^{(m)} \rangle &= E\{S_t^{(m)}|\theta, \{\mathbf{y}_t\}\} \qquad (7) \\ \langle S_{t-1}^{(m)} S_t^{(m)'} \rangle &= E\{S_{t-1}^{(m)} S_t^{(m)'}|\theta, \{\mathbf{y}_t\}\} \qquad (8) \\ \langle S_t^{(m)} S_t^{(n)'} \rangle &= E\{S_t^{(m)} S_t^{(n)'}|\theta, \{\mathbf{y}_t\}\} \qquad (9) \\ \textbf{Complete Dataset:} &= \{\{\mathbf{y}_t\}, \{\langle S_t^{(m)} \rangle\}, \{\langle S_{t-1}^{(m)} S_t^{(m)'} \rangle\}, \{\langle S_t^{(m)} S_t^{(n)'} \rangle\}\} \qquad (10) \end{aligned}$$

The expectation of (7) corresponds to the posterior state occupation probabilities of a first order HMM. The expectation of (8) is a $K \times K$ matrix which corresponds to the state transition probabilities of a first order HMM. The expectation of (9) is also a $K \times K$ matrix that arises from the data conditional dependency between hidden states at the same time unit, and has no equivalent posterior in a first order HMM.

3.2 An Expectation Maximization Algorithm for Learning the FHMM

[19] [39] [42] To estimate the FHMM model we must first define an error function which measures the difference between the observation distribution generated by our model $P(\{\mathbf{y}_t\}|\theta)$ and the actual (unknown) observation distribution $\hat{P}(\{\mathbf{y}_t\})$. We choose the Kullback-Leibler (KL) distance function [58]:

$$\begin{aligned} \mathcal{E}_0(\theta) &= \sum_{\{\mathbf{y}_t\}} \hat{P}(\{\mathbf{y}_t\}) \, log \left[\frac{\hat{P}(\{\mathbf{y}_t\})}{P(\{\mathbf{y}_t\}|\theta)} \right] \\ &= -\sum_{\{\mathbf{y}_t\}} \hat{P}(\{\mathbf{y}_t\}) \, log[P(\{\mathbf{y}_t\}|\theta)] + \sum_{\{\mathbf{y}_t\}} \hat{P}(\{\mathbf{y}_t\}) \, log[\hat{P}(\{\mathbf{y}_t\})] \\ &= -E\left\{log \, P(\{\mathbf{y}_t\}|\theta)\right\} - H_{\hat{P}} \qquad (11) \end{aligned}$$

where the operator E performs averaging over the observed $\{\mathbf{y}_t\}$. As is well known the KL distance $\mathcal{E}_0$ is non-negative and vanishes when $P(\{\mathbf{y}_t\}|\theta) = \hat{P}(\{\mathbf{y}_t\})$.

The second term $H_{\hat{P}}$ of equation (11) is the entropy of the observations. Since it is independent of the model parameters θ it will henceforth be dropped. The first term is a functions of the negative log-likelihood of the observations given the model parameters θ. Minimizing the error is therefore equivalent to minimizing the negative log-likelihood of the observations given the model parameters; i.e. maximizing the likelihood of the data with respect to the model.

$$\begin{aligned} &\text{Minimize:} \quad \mathcal{E}(\theta) = E\{-\, log\ P(\{\mathbf{y}_t\}|\theta)\} \\ &\text{Maximize:} \quad \mathcal{L}(\theta) = E\{log\ P(\{\mathbf{y}_t\}|\theta)\} \end{aligned} \tag{12}$$

The Expectation Maximization (EM) algorithm is an iterative algorithm to maximize the expected log-likelihood of the observed data with respect to the parameters of the model describing the data. It is obtained by considering, in addition to the expected log likelihood $E[log\ P(\{\mathbf{y}_t\}|\theta)]$ of the observed data, the expected log likelihood of the 'complete' data $E[log\ P(\{S_t, \mathbf{y}_t\}, |\theta)]$ composed of both the observed and the 'missing' data for the latent states. For the FHMM the complete data is given by (10). Each iteration is defined in two steps:

1. **Expectation (E) step:**
   Calculate the expected value of the complete data likelihood, given the observed data and the current model, where θ is the set of current model parameters and $\hat{\theta}$ is the set of new model parameters.
   $$\mathcal{F}(\theta, \hat{\theta}) = E\left\{ logP(\{S_t, \mathbf{y}_t\}|\hat{\theta}) \right\} - \mathcal{F}_H(\theta) \tag{13}$$
   $\mathcal{F}_H(\theta)$ is the entropy of the posterior w.r.t to θ , the current model set parameters considered.

2. **Maximization (M) step:**
   Maximize $\mathcal{F}(\theta, \hat{\theta})$ with respect to the model to obtain the new model parameters:
   $$\hat{\theta} = \arg\max_{\tilde{\theta}} \mathcal{F}(\theta, \tilde{\theta}) \tag{14}$$

To develop an EM algorithm for learning the FHMM we first prove that the likelihood $\mathcal{L}$ to be maximized given by (12) is bounded from below by $\mathcal{F}$ from (13), and that any arbitrary distribution over the latent variables $Q(\{S_t\}|\theta)$ can be used to define this lower bound.

$$\begin{aligned} \text{Maximize:} \quad \mathcal{L}(\hat{\theta}) &= E\{log\ P(\{\mathbf{y}_t\}|\hat{\theta})\} \\ &= E\left\{ log \sum_{\{S_t\}} P(\{S_t, \mathbf{y}_t\}|\hat{\theta}) \right\} \\ &= E\left\{ log \sum_{\{S_t\}} Q(\{S_t\}|\theta)\ \frac{P(\{S_t, \mathbf{y}_t\}|\hat{\theta})}{Q(\{S_t\}|\theta)} \right\} \\ &\geq E\left\{ \sum_{\{S_t\}} Q(\{S_t\}|\theta)\ log \left[\frac{P(\{S_t, \mathbf{y}_t\}|\hat{\theta})}{Q(\{S_t\}|\theta)} \right] \right\} \equiv \mathcal{F}(\theta, \hat{\theta}) \end{aligned} \tag{15}$$

The last step giving inequality (15) follows from Jensen's inequality since $log(x)$ is a convex function.

Jensen's inequality [58]:

$$\text{If } f \text{ is a convex function: } f\left(\sum_i p_i x_i \right) \geq \sum_i p_i f(x_i) \quad \text{where} \quad \sum_i p_i = 1$$

Result (15) proves that any distribution $Q(\{S_t\}|\theta)$ over the latent variables can be used to define a lower bound $\mathcal{F}$ on the log likelihood to be maximized.

We choose distribution Q to be the data conditional posterior distribution over the latent variables of the current model.

$$Q(\{S_t\}|\theta) = P(\{S_t\}|\{\mathbf{y}_t\}, \theta) \tag{16}$$

When $\hat{\theta} = \theta$ this choice makes (15) an equality ...

$$\begin{aligned} \mathcal{L}(\theta) &= \mathcal{F}(\theta, \theta) \\ &\leq \mathcal{F}(\theta, \hat{\theta}) \quad \text{according to (14)} \\ &\leq \mathcal{L}(\hat{\theta}) \end{aligned}$$

... and ensures that each EM step does not decrease the likelihood of the observed data.

A major part of the E-step is the estimation of the posterior probabilities of the latent variables given the observation, as defined by (16). This is defined as the **Inference Problem**.

There are 2 parts to learning the parameters for the problem. Learning the optimal structure for the problem is defined as the **Structure Learning Problem** or the **Model Selection problem**, which consists of defining parameters K, T and M for the problem at hand. Given the suitable structure for the problem, the **Parameter Learning Problem** consists of learning the FHMM parameters given in (6) according to the M-step.

3.3 The M-step for the FHMM is Tractable

[3] [4] The M-step consists of learning the parameters given in (6) according to (14), given the complete dataset defined by (10). The parameters are derived from setting the derivative of $\mathcal{F}(\theta, \hat{\theta})$ with respect to each parameter to zero. However, only the first term of $\mathcal{F}$ in (13) is considered as the second term is not a function of the new parameters. We expand $\mathcal{F}$ in terms of equations (1) through (4).

$$\begin{aligned} \mathcal{F} &= E\left\{log\ P(\{S_t, \mathbf{y}_t\}|\hat{\theta})\right\} \\ &= E\left\{log\ \left[P(S_1)\ \prod_{t=2}^{T} P(S_t|S_{t-1})\ \prod_{t=1}^{T} P(\mathbf{y}_t|S_t)\right]\right\} \qquad (17) \\ &= E\left\{\sum_{m=1}^{M} S_1^{(m)'}\ logP(S_1^{(m)})\right\} + E\left\{\sum_{t=2}^{T}\sum_{m=1}^{M} S_t^{(m)'}(logP(S_t^{(m)}|S_{t-1}^{(m)}))S_{t-1}^{(m)}\right\} \\ &\quad + E\left\{\sum_{t=1}^{T} logP(\mathbf{y}_t|S_t)\right\} \\ &= \sum_{m=1}^{M} \langle S_1^{(m)'}\rangle\ logP(S_1^{(m)})\ +\ \sum_{t=2}^{T}\sum_{m=1}^{M} Tr\{\langle S_{t-1}^{(m)} S_t^{(m)'}\rangle\ logP(S_t^{(m)}|S_{t-1}^{(m)})\} \\ &\quad -\frac{1}{2}\sum_{t=1}^{T}\left\{\mathbf{y}_t'\Sigma^{-1}\mathbf{y}_t - 2\sum_{m=1}^{M}\mathbf{y}_t'\Sigma^{-1}W^{(m)}\langle S_t^{(m)}\rangle + \sum_{m=1}^{M}\sum_{n=1}^{M} Tr\{W^{(m)'}\Sigma^{-1}W^{(n)}\langle S_t^{(n)} S_t^{(m)'}\rangle\}\right\} \\ &\quad -\frac{1}{2}\,log(2\pi)^{DT}\ +\ \frac{1}{2}\,log|\Sigma^{-1}|^T\ -\ logZ \qquad (18) \end{aligned}$$

where Z is the normalization term independent of the states and the observations that ensures all probabilities sum to 1.

The maximum values of the parameters are estimated by setting the derivatives of $\mathcal{F}$ with respect to each of the parameters $\theta = \{W^{(m)}, \pi^{(m)}, P^{(m)}, \Sigma\}$ to 0.

$$\frac{\partial \mathcal{F}}{\partial W^{(m)}} = \sum_{t=1}^{T} \langle S_t^{(m)'} \rangle \mathbf{y}_t \; - \; \sum_{t=1}^{T} \sum_{n=1}^{M} W^{(n)'} \langle S_t^{(n)} S_t^{(m)'} \rangle = 0 \tag{19}$$

Let S_t be the $MK \times 1$ vector given by concatenating the M vectors $S_t^{(m)}$, and W be the $D \times MK$ matrix from concatenating the M matrices side by side.

$$W^{new} \; = \; \left(\sum_{t=1}^{T} \mathbf{y}_t \langle S_t^{'} \rangle \right) \left(\sum_{t=1}^{T} \langle S_t S_t^{'} \rangle \right)^{\dagger}$$

where $\dagger$ is the Moore-Penrose pseudo inverse $A^{\dagger} = A^{'}(AA^{'})^{-1}$.

This operation is at most $O(M^3K^3)$, complexity arising from the matrix inversion or multiplication for calculating the pseudo-inverse.

Taking the derivative with respect to Σ^{-1}:

$$\begin{aligned} \frac{\partial \mathcal{F}}{\partial \Sigma^{-1}} \quad = \quad & -\frac{1}{2} \sum_{t=1}^{T} \mathbf{y}_t \mathbf{y}_t^{'} + \sum_{t=1}^{T} \sum_{m=1}^{M} W^{(m)} \langle S_t^{(m)} \rangle \mathbf{y}_t^{'} \\ & -\frac{1}{2} \sum_{t=1}^{T} \sum_{n=1}^{M} \sum_{m=1}^{M} W^{(n)'} \langle S_t^{(n)} S_t^{(m)'} \rangle W^{(m)} + \frac{T}{2} \Sigma = 0 \end{aligned} \tag{20}$$

Substituting after rearranging from (19),

$$\sum_{t=1}^{T} \sum_{n=1}^{M} \sum_{m=1}^{M} W^{(n)'} \langle S_t^{(n)} S_t^{(m)'} \rangle W^{(m)} = \sum_{t=1}^{T} \sum_{m=1}^{M} W^{(m)} \langle S_t^{(m)} \rangle \mathbf{y}_t^{'}$$

$$\Sigma^{new} = \frac{1}{T} \sum_{t=1}^{T} \mathbf{y}_t \mathbf{y}_t^{'} - \frac{1}{T} \sum_{t=1}^{T} \sum_{m=1}^{M} W^{(m)} \langle S_t^{(m)} \rangle \mathbf{y}_t^{'} \tag{21}$$

The complexity of the above operation is $O(TMDK^2)$, arising from the calculation of the second term.

Taking the derivative, with respect to $P(S_1^{(m)})$, of $\mathcal{F}$ subject to the constraint the probabilities sum to 1, $\sum_{m=1}^{M} P(S_1^{(m)}) = 1$,

$$\frac{\langle S_1^{(m)} \rangle}{P(S_t^{(m)})} - 1 = 0 \tag{22}$$

$$\pi^{(m)} = P(S_t^{(m)}) = \langle S_1^{(m)} \rangle \tag{23}$$

This operation is trivial.

(**Note**: Need to understand $P(S_t^{(m)} | S_{t-1}^{(m)})$ derivation.)

The parameters are estimated in terms of the posteriors given by (7), 8) and (9):

- $\langle S_t^{(m)} \rangle$, the source posterior probabilities
- $\langle S_{t-1}^{(m)} S_t^{(m)'} \rangle$, the matrix giving the posterior transition probabilities between two states during two consecutive time units, and

- $\langle S_t^{(n)} S_t^{(m)'} \rangle$, the matrix giving the posterior transition probabilities between two states at the same time unit.

Given these posteriors, all of the parameter derivations above are simple and tractable.

These posteriors are estimated during the E-step of the algorithm and their estimation is the **Inference Problem** of learning the FHMM.

3.4 The Exact Inference Problem for the FHMM is Intractable

In (15) we proved that *any* distribution over the latent variables $Q(\{S_t\}|\theta)$ can be used to define the lower bound $\mathcal{F}(\theta, \hat{\theta})$ of the log likelihood $\mathcal{L}(\hat{\theta})$.

We solve the Inference Problem for the model by minimizing the divergence $\mathcal{D}$ between the actual expected log likelihood and the lower bound defined by (15) according to the approximating distribution Q. We thereby make the lower bound as tight as possible; i.e. we minimize the divergence between the true expected log likelihood and our model's approximation of it. This divergence is the difference between the LHS and the RHS of the inequality (15).

$$
\begin{aligned}
\text{Minimize: } \mathcal{D} &= \mathcal{L}(\hat{\theta}) - \mathcal{F}(\theta, \hat{\theta}) \\
&= E\left\{ logP(\{\mathbf{y}_t\}|\hat{\theta}) - \sum_{\{S_t\}} Q(\{S_t\}|\theta)\ log\left[\frac{P(\{S_t, \mathbf{y}_t\}|\hat{\theta})}{Q(\{S_t\}|\theta)}\right]\right\} \\
&= E\left\{ \sum_{\{S_t\}} Q(\{S_t\}|\theta) logP(\{\mathbf{y}_t\}|\hat{\theta}) - \sum_{\{S_t\}} Q(\{S_t\}|\theta)\ log\left[\frac{P(\{S_t, \mathbf{y}_t\}|\hat{\theta})}{Q(\{S_t\}|\theta)}\right]\right\} \\
&= E\left\{ \sum_{\{S_t\}} Q(\{S_t\}|\theta)\ log\left[\frac{P(\{\mathbf{y}_t\}|\hat{\theta})\ Q(\{S_t\})}{P(\{S_t, \mathbf{y}_t\}|\hat{\theta})}\right]\right\} \\
&= E\left\{ \sum_{\{S_t\}} Q(\{S_t\}|\theta)\ log\left[\frac{Q(\{S_t\}|\theta)}{P(\{S_t\}|\{\mathbf{y}_t\}, \hat{\theta})}\right]\right\} = KLD(Q||\hat{P}) \qquad (24)
\end{aligned}
$$

As shown in (24) this is exactly the Kullback-Leibler divergence between the expected data conditional posterior over the latent states of the new model set $P(\{S_t\}|\{\mathbf{y}_t\}, \hat{\theta})$, and the approximating distribution $Q(\{S_t\}|\theta)$ we define over the latent states of the current model set.

Hence the Inference Problem becomes a constrained minimization problem subject to the constraints that all probabilities concerned sum to 1. The problem must solve for the unknowns which are the posteriors defined by (7), (8) and (9).

For Exact Inference we pick Q to be the data dependent posterior over the hidden states of the current model.

$$Q(\{S_t\}|\theta) = P(\{S_t\}|\{\mathbf{y}_t\}, \theta) \qquad (25)$$

This choice makes the constrained minimization problem for the FHMM intractable.

$$
\begin{aligned}
KL(Q||\hat{P}) &= E\left\{log\, Q(\{S_t\}|\theta) - log\, P(\{S_t\}|\{\mathbf{y}_t\},\hat{\theta})\right\} \\
&= E\left\{log\, P(\{S_t\}|\{\mathbf{y}_t\},\theta) - log\left[\frac{P(\{S_t,\mathbf{y}_t\}|\hat{\theta})}{P(\{\mathbf{y}_t\}|\hat{\theta})}\right]\right\} \\
&\propto E\left\{log\left[\frac{P(\{S_t,\mathbf{y}_t\}|\theta)}{P(\{\mathbf{y}_t\}|\theta)}\right] - log\, P(\{S_t,\mathbf{y}_t\}|\hat{\theta})\right\} \\
&= E\left\{log\left[P(\{S_t,\mathbf{y}_t\}|\theta)\right] - log\left[P(\{S_t,\mathbf{y}_t\}|\hat{\theta})\right] - log\left[P(\{\mathbf{y}_t\}|\theta)\right]\right\}
\end{aligned}
$$

The proportionality of the third step follows from the fact that $\hat{P}(\{\mathbf{y}_t\}) = P(\{\mathbf{y}_t\}|\hat{\theta})$ is independent of the unknown posteriors and can be disregarded for the purpose of the minimization.

Consider the estimation of the third expression of the equation:

$$
\begin{aligned}
E\left\{log\left[P(\{\mathbf{y}_t\}|\theta)\right]\right\} &= E\left\{log\left[\prod_{t=1}^{T}\sum_{c=1}^{K^M} p_c\,\mathcal{N}(\mathbf{y}_t : \mu_{tc}, \Sigma)\right]\right\} \\
&= E\left\{\sum_{t=1}^{T} log\left[\sum_{c=1}^{K^M} p_c\,\mathcal{N}(\mathbf{y}_t : \mu_{tc}, \Sigma)\right]\right\}
\end{aligned}
$$

This derives from the probability model given according to (5). The mean μ_{tc} is formed by selecting a column from each of the $D \times K$ mixing matrices $W^{(m)}$; hence μ_{tc} can be chosen in K^M different ways in time $O(K^M)$. It takes time $O(MK)$ to sum the M chosen $K \times 1$ columns to form each mean. Hence the above derivation is intractable and run in time $O(TMK^{M+1})$. Therefore the E-step for exact inference runs in time order $O(TMK^{M+1})$ and is intractable for $M > 2$.

3.5 The Mean Field Approximation

[3] [4] In (15) we proved that <u>*any*</u> distribution over the latent variables $Q(\{S_t\}|\theta)$ can be used to define the lower bound $\mathcal{F}(\theta,\hat{\theta})$ of the log likelihood $\mathcal{L}$.

The intractability of the Inference Problem arises from our choice of Q to be the exact posterior distribution $P(\{S_t\}|\{\mathbf{y}_t\},\theta)$ over the hidden variables of the model from the previous iteration.

In the Mean Field approximation we choose Q to be a distribution over the latent variables of the model from the previous iteration *that is tractable*. For the Mean Field distribution we approximate that:

i. the hidden states are independent of the observations,

$$P(S_t^{(m)}|\{\mathbf{y}_t^{(m)}\},\vartheta) = P(S_t^{(m)}|\vartheta) \tag{26}$$

ii. the hidden states are independent of any other hidden states.

$$
\begin{aligned}
P(S_t^{(m)}|\{S_{t-1}^{(m)}\},\vartheta) &= P(S_t^{(m)}|\vartheta) &(27)\\
P(S_t^{(m)}|S_t^{(n)},\vartheta) &= P(S_t^{(m)}|\vartheta) \quad , n \neq m &(28)
\end{aligned}
$$

The Mean Field distribution Q is shown by figure 7 and has the probability distribution:

$$Q(\{S_t\}|\theta) = P(\{S_t\}|\vartheta) = \prod_{t=1}^{T}\prod_{m=1}^{M} P(S_t^{(m)}|\vartheta_t^{(m)}) \tag{29}$$

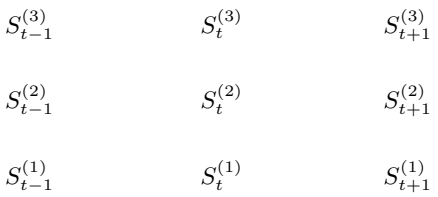


Figure 7: The Mean Field Approximation

The intuition behind the Mean Field approximation is that in a dense graph each node is subject to influences from many other nodes. Thus to the extent that each influence is weak and the total influence is roughly additive, each node should roughly be characterized by its mean value; i.e. each node fluctuates independently about its mean value.[7]

The parameters $\vartheta = \{\vartheta_t^{(m)}\}$ are defined as the *variational parameters* of the model. They are the means of the state variables $\{S_t^{(m)}\}$, hence the term Mean Field. Each $S_t^{(m)}$ is a K dimensional vector with a 1 in the k^{th} position and 0 elsewhere, that defines which of the K states the variable occupies at time t. Each $\vartheta_t^{m)}$ is the K dimensional mean vector that defines the occupation probabilities of each of the K states at time t. The elements of the vector $\vartheta_t^{(m)}$ therefore define the state occupation probabilities of the multinomial variable $S_t^{(m)}$ under the distribution Q.

$$P(S_t^{(m)}|\vartheta) = \prod_{k=1}^{K} \left(\vartheta_{t,k}^{(m)}\right)^{S_{t,k}^{(m)}} \quad \text{where} \quad \sum_{k=1}^{K} S_{t,k}^{(m)} = 1 \tag{30}$$

A useful result that derives from this approximation is:

$$log\left[P(\{S_t^{(m)}\}|\vartheta)\right] \;=\; \sum_{t=1}^{T}\sum_{m=1}^{M}\sum_{k=1}^{K} S_{t,k}^{(m)}\, log\, \vartheta_{t,k}^{(m)} \;=\; \sum_{t=1}^{T}\sum_{m=1}^{M} S_t^{(m)'}\, log\, \vartheta_t^{(m)} \tag{31}$$

From the definition of the variational parameters:

$$\begin{aligned}
\langle S_t^{(m)}\rangle &= \vartheta_t^{(m)} && (32)\\
\langle S_{t-1}^{(m)} S_t^{(m)'}\rangle &= \vartheta_{t-1}^{(m)}\vartheta_t^{(m)'} && (33)\\
\langle S_t^{(m)} S_t^{(n)'}\rangle &= \vartheta_t^{(m)}\vartheta_t^{(n)'} \quad \text{, for } n \neq m && (34)\\
&= \text{diag}\{\vartheta_t^{(m)}\} \text{ , for } n = m && (35)
\end{aligned}$$

$$\begin{aligned}
KL(Q||\hat{P}) &= \mathcal{E}\left\{log\, Q(\{S_t\}|\theta) - log\, P(\{S_t\}|\{\mathbf{y}_t\},\hat{\theta})\right\}\\
&= \mathcal{E}\left\{log\, Q(\{S_t\}|\theta) - log\left[\frac{P(\{S_t,\mathbf{y}_t\}|\hat{\theta})}{P(\{\mathbf{y}_t\}|\hat{\theta})}\right]\right\}\\
&\propto \mathcal{E}\left\{log\, Q(\{S_t\}|\theta) - log\, P(\{S_t,\mathbf{y}_t\}|\hat{\theta})\right\}\\
&= \mathcal{E}\left\{log\left[\frac{1}{Z_Q}\, exp\{-H_Q(\{S_t\})\}\right] - log\left[\frac{1}{Z_{\hat{P}}}\, exp\{-H_{\hat{P}}(\{S_t,\mathbf{y}_t\})\}\right]\right\}\\
&= \langle H_{\hat{P}}\rangle \;-\; \langle H_Q\rangle \;+\; log\, Z_{\hat{P}} \;-\; log\, Z_Q
\end{aligned} \tag{36}$$

To determine $\vartheta_t^{(m)}$ we minimize the divergence given in (36).

$$
\begin{aligned}
H_{\hat{P}} &= \frac{1}{2}\sum_{t=1}^{T}\left(\mathbf{y}_t - \sum_{m=1}^{M} W^{(m)} S_t^{(m)}\right)' \Sigma^{-1}\left(\mathbf{y}_t - \sum_{m=1}^{M} W^{(m)} S_t^{(m)}\right) \\
&\quad - \sum_{m=1}^{M} S_1^{(m)'} log\, \pi^{(m)} - \sum_{t=1}^{T}\sum_{m=1}^{M} S_t^{(m)'} (log\, P^{(m)}) S_{t-1}^{(m)} \\
H_Q &= -\sum_{t=1}^{T}\sum_{m=1}^{M} S_t^{(m)'}\, log\, \vartheta_t^{(m)}
\end{aligned}
$$

$$
\begin{aligned}
KD(\hat{P}||Q) &= \langle H_{\hat{P}} \rangle - \langle H_Q \rangle + log\, Z_{\hat{P}} - log\, Z_Q \\
&= \frac{1}{2}\sum_{t=1}^{T}\left[\mathbf{y}_t' \Sigma^{-1} \mathbf{y}_t - 2\sum_{m=1}^{M} W^{(m)'} \Sigma^{-1} \mathbf{y}_t\, \vartheta_t^{(m)}\right] + \\
&\quad \frac{1}{2}\sum_{t=1}^{T}\left[\sum_{m=1}^{M}\sum_{n\neq m} tr\{W^{(m)'} \Sigma^{-1} W^{(n)} \vartheta_t^{(n)} \vartheta_t^{(m)'}\} + \sum_{m=1}^{M} tr\{W^{(m)'} \Sigma^{-1} W^{(m)}\, diag\{\vartheta_t^{(m)}\}\}\right] \\
&\quad - \sum_{m=1}^{M} \vartheta_1^{(m)'} log\, \pi^{(m)} - \sum_{t=2}^{T}\sum_{m=1}^{M} tr\{\vartheta_{t-1}^{(m)} \vartheta_t^{(m)'} (log\, P^{(m)})\} \\
&\quad + \sum_{t=1}^{T}\sum_{m=1}^{M} \vartheta_t^{(m)'}\, log\, \vartheta_t^{(m)} + log\, Z_{\hat{P}} - log\, Z_Q
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial KD}{\partial\, \vartheta_t^{(m)}} &= -W^{(m)'} \Sigma^{-1} \mathbf{y}_t + \sum_{n\neq m} W^{(m)'} \Sigma^{-1} W^{(n)} \vartheta_t^{(n)} + diag\{W^{(m)'} \Sigma^{-1} W^{(m)}\} \\
&\quad - (log\, P^{(m)})' \vartheta_{t-1}^{(m)} - (log\, P^{(m)})' \vartheta_{t+1}^{(m)} + log\, \vartheta_t^{(m)} + C
\end{aligned}
$$

where C is the constant that ensures the probabilities $\vartheta_t^{(m)}$ sum to 1.

Setting $\frac{\partial KD}{\partial\, \vartheta_t^{(m)}}$ equal to 0:

$$
\begin{aligned}
\vartheta_t^{(m)new} &= \psi\left[(log\, P^{(m)})'(\vartheta_{t-1}^{(m)} + \vartheta_{t+1}^{(m)}) + W^{(m)'} \Sigma^{-1} \tilde{\mathbf{y}}_t - diag\{W^{(m)'} \Sigma^{-1} W^{(m)}\}\right] \quad (37) \\
\text{where} \quad \tilde{Y} &= \mathbf{y}_t - \sum_{n\neq m} \Sigma^{-1} W^{(n)} \vartheta_t^{(n)}
\end{aligned}
$$

The function ψ maps each element A_i of vector A to an element in vector B where each B_i is the exponent of A_i, normalized to ensure that the vector elements in B sum to 1.

$$
B_i = \frac{exp\{A_i\}}{\sum_i \exp\{A_i\}}
$$

Estimation of $\vartheta_t^{(m)new}$ is tractable (Note: give time order) and renders the derivation of the posterior statistics of (32) through (35) tractable. The statistics can then be used to estimate the parameters in the M-step tractably.

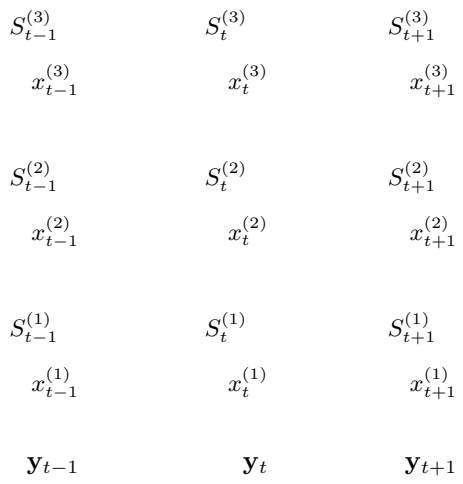


Figure 8: The FHMM Gaussian BSS Model

4 Learning the FHMM Modelling Continuous Non-Gaussian and Gaussian Source and Sensor Signals

[19] [20] [21]The sources are modelled by continuous distributions, where a source can assume any distribution. An arbitrary distribution can be approximated by a Mixture of Gaussian's, the accuracy increasing with an increase in the number of mixture components. The situation where the sources are Gaussian is a special case of the model when the number of components (or states in each stream) is 1. An analysis based on a dynamic extension of Independent Factor Analysis is presented. The initial model is derived in the time domain and factors for adaptation to the frequency and cepstral domains are derived. The analysis derives from a signal processing perspective of Blind Source Separation (BSS) using ICA, as well as from an applied statistical perspective of IFA. The notation and terminology is therefore adapted to correspond to the increasingly Signal Processing based origins of model analysis.

4.1 The FHMM Probability Model for Gaussian Source Separation

The properties of the FHMM model, shown in figure 8 are summarized as follows for the case where the sources are modelled as single Gaussian's. The model is presented in the time domain in the first instance, and will be adapted to the frequency and cepstral domains at a later stage.

- The models is a time series model, for time instances $t = 1 \dots T$.
- There are $m = 1 \dots M$ sources. The sources at each time instance t are represented by M hidden state variables $\mathbf{s_t} = (s_t^{(1)}, \dots, s_t^{(m)}, \dots, s_t^{(M)})$. Source m is in state $s_t^{(m)}$ at time t. The collective hidden state variable for a time instance is given by $\mathbf{s_t}$.

  There are $K^{(m)}$ possible states generating the source signal in each stream. These states can be represented by a $K^{(m)}$ length vector $S_t^{(m)}$, with 1 in the occupied state at time t and 0 elsewhere. At time instance t the process of generating the signal in stream m can be in each state s with state occupation probability $\gamma_t^{(m)}(s)$. The state occupation probabilities in stream m at time t can be represented by the $K^{(m)}$ length vector $\langle S_t^{(m)} \rangle$ where $\langle \cdot \rangle$ denotes taking the expectation over T time instances.
- **Individual Source Model:** The signal $x_t^{(m)}$ of hidden state $s_t^{(m)}$ is generated by sampling from a continuous distribution. Each source signal, given a state, is drawn from a one

dimensional Gaussian with $\mu_t^{(m)}$ mean and $\nu_t^{(m)}$ variance:

$$\begin{aligned} P(x_t^{(m)}|s_t^{(m)}, \theta_t^{(m)}) &= \mathcal{N}\left(x_t^{(m)} : \mu_t^{(m)}, \nu_t^{(m)}\right) \\ \text{parameters } \theta_t^{(m)} &= \left\{\mu_t^{(m)}, \nu_t^{(m)}\right\} \end{aligned} \tag{38}$$

This is a generative model of the source signal $x_t^{(m)}$ in the one dimensional space. To generate the source signal $x_t^{(m)}$ given state $s_t^{(m)}$ we sample from the corresponding Gaussian $P(x_t^{(m)}|s_t^{(m)}) = \mathcal{N}\left(x_t^{(m)} : \mu_t^{(m)}, \nu_t^{(m)}\right)$.

At time t, given the state occupation probabilities in stream m, we generate the source signal by choosing the state s with its state occupation probability, and sampling from the corresponding Gaussian.

$$P(x_t^{(m)}|\theta^{(m)}) = \sum_{s=1}^{K^{(m)}} \gamma_t^{(m)}(s)\, \mathcal{N}\left(x_t^{(m)} : \mu_t^{(m)}, \nu_t^{(m)}\right) \tag{39}$$

Hence the individual source model at time t is a one dimensional mixture of Gaussian's.

- **M-dimensional Source Model:** In the M dimensional space the source signals at time t form an M length vector $\mathbf{x_t} = (x_t^{(1)}, x_t^{(2)}, \ldots, x_t^{(M)})$.

  The M dimensional sources at time t are represented by a collective source state $\mathbf{s}_t = (s_t^{(1)}, s_t^{(2)}, \ldots, s_t^{(M)})$.

  The collective state has the concatenated mean: $\mu_\mathbf{t} = \left(\mu_t^{(1)}, \mu_t^{(2)}, \ldots, \mu_t^{(M)}\right)$

  and the covariance: $\mathbf{V}_t = \text{diag}\left\{\nu_t^{(1)}, \nu_t^{(2)}, \ldots, \nu_t^{(M)}\right\}$

  Hence the M dimensional source model at time t, given the occupied states, is an M dimensional Gaussian:

$$\begin{aligned} P(\mathbf{x}_t|\mathbf{s}_t, \theta_t^{source}) &= \mathcal{N}\left(\mathbf{x}_t : \mu_t, \mathbf{V}_t\right) \\ \text{parameters } \theta_t^{(source)} &= \{\mu_t, \mathbf{V}_t\} \end{aligned} \tag{40}$$

  Picking a state from the $K^{(m)}$ states in each stream, there are $K = \prod_m K^{(m)}$ ways a collective state $\mathbf{s_t}$ can be formed, each occupied with the collective occupation probability $\gamma_t(\mathbf{s}_t) = \prod_m \gamma_t^{(m)}(s^{(m)})$, where $s^{(m)}$ indexes the state occupied in stream m. Hence the M dimensional source model at time t is a co-adaptive Factorial Mixture of Gaussians:

$$P(\mathbf{x}_t|\theta) = \sum_{\mathbf{s}_t=1}^{K} \gamma_t(\mathbf{s}_t)\, \mathcal{N}\left(\mathbf{x}_t : \mu_t, \mathbf{V}_t\right) \tag{41}$$

- **Mixing:** Mixing is considered instantaneous: i.e. only the source signals at time t contribute to the sensor signal at time t.

  The source and sensor signals at time t are coupled according to a $D \times M$ mixing matrix $\mathbf{H} = [h_d^{(m)}] = [\mathbf{h}_1, \ldots, \mathbf{h}_D]^T$ and D dimensional mixing noise vector $\mathbf{u_t}$, where D is the number of sensor signals:

$$\mathbf{y}_t = \mathbf{H}\mathbf{x}_t + \mathbf{u}_t \tag{42}$$

Consider the d^{th} sensor signal $y_{t,d}$ where $\mathbf{y}_t = (y_{t,1}, \ldots, y_{t,D})$:

$$y_{t,d} = \sum_{m=1}^{M} h_d^{(m)} * x_t^{(m)} + u_{t,d} \tag{43}$$

The convolutions of M mutually uncorrelated source signals with the impulse response $h_d^{(m)}$ of their corresponding channels, are added together. Hence, $h_d^{(m)}$ can be interpreted as the impulse response of the channel from source m to sensor d. A further additive noise signal is added to this to give the sensor signal.

- **Unmixing:** Assuming invertible $\mathbf{H}$ and no mixing noise,

$$\mathbf{x}_t = \mathbf{G}\mathbf{y}_t \tag{44}$$
$$\text{where } \mathbf{G} = \mathbf{H}^{-1} \tag{45}$$

In reverse, the sensor signals are factored into the M ($\leq D$) most important independent source signals, convolved with the impulse response of the respective channel to each sensor. This is the *deterministic* dynamic processes generating the observations in the model.

The remaining (unimportant) factors from the decomposition are collectively modelled together with stochastic noise as residual additive noise. This is the *chaotic* portion of the dynamic processes contributing to generating the observations in the model.

This factoring, or Blind Source Separation, is performed using ICA/IFA. The estimation performs a source separation and deconvolution process, estimating the important source signals, the impulse responses of the channels and any residual additive noise.

- **Mixing Noise:** The mixing noise is considered to be white noise, modelled as Gaussian with zero mean:

$$P(\mathbf{u}_t) = \mathcal{N}(\mathbf{u}_t : 0, \mathbf{\Lambda}_t) \tag{46}$$

A diagonal covariance denotes that the remaining factors of the decomposition are uncorrelated, as is the case with decomposition using PPCA/FA. The analysis becomes increasingly complex when moving from an isotropic noise model (with covariance of the form $\sigma^2\mathbf{I}$), to a general non-isotropic noise model with a full covariance.

- **Sensor Model:** From the source, mixing and noise models above, the sensor model conditional on the sources is:

$$P(\mathbf{y}_t|\mathbf{x}_t, \theta_t) = \mathcal{N}(\mathbf{y}_t : \mathbf{H}\mathbf{x}_t, \mathbf{\Lambda}_t) \tag{47}$$
$$\text{parameters } \theta_t = \{\mu_t, \mathbf{V}_t, \mathbf{\Lambda}_t\}$$

It can be shown that the sensor model, conditional on the state at time t, is also Gaussian.

$$P(\mathbf{y}_t|\mathbf{s}_t, \theta_t) = \mathcal{N}(\mathbf{y}_t : \mathbf{H}\mu_t,\ \mathbf{H}'\mathbf{V}_t\mathbf{H} + \mathbf{\Lambda}_t) \tag{48}$$

Given the state occupation probabilities at time t for states generating the sources,

$$P(\mathbf{s}_t) = \gamma_t(\mathbf{s}_t) = \prod_{m=1}^{M} \gamma_t^{(m)}(s_t^{(m)}) \tag{49}$$

$$P(\mathbf{y}_t,\ \mathbf{s}_t|\theta_t) = \gamma_t(\mathbf{s}_t)\,\mathcal{N}(\mathbf{y}_t : \mathbf{H}\mu_t,\ \mathbf{H}'\mathbf{V}_t\mathbf{H} + \mathbf{\Lambda}_t) \tag{50}$$

Summing over all possible state combinations of states in each source stream, it can be shown that the sensor model at time t is a Mixture of Gaussian's. If there are $K^{(m)}$ states for each source stream, there are $\prod_{m=1}^{M} K^{(m)}$ possible state combinations.

$$P(\mathbf{y}_t|\theta_t) = \sum_{\mathbf{s}_t} \gamma_t(\mathbf{s}_t)\, \mathcal{N}(\mathbf{y}_t : \mathbf{H}\mu_t,\ \mathbf{H}^{'}\mathbf{V}_t\mathbf{H} + \mathbf{\Lambda}_t) \tag{51}$$

$$P(\{\mathbf{y}_t\}|\theta) = \prod_{t=1}^{T} \sum_{\mathbf{s}_t} \gamma_t(\mathbf{s}_t)\, \mathcal{N}(\mathbf{y}_t : \mathbf{H}\mu_t,\ \mathbf{H}^{'}\mathbf{V}_t\mathbf{H} + \mathbf{\Lambda}_t) \tag{52}$$

- The hidden states observe the Markov property such that each state variable is independent of the other state variables, given the preceding state variable.

$$P(\mathbf{s}_t|\mathbf{s}_{t-1}, \theta) = \prod_{m=1}^{M} P(S_t^{(m)}|S_{t-1}^{(m)}) \tag{53}$$

$$P(\{\mathbf{y}_t, \mathbf{x}_t, \mathbf{s}_t\}|\theta) = \prod_{t=2}^{T} P(\mathbf{s}_t|\mathbf{s}_{t-1}) \prod_{t=1}^{T} P(\mathbf{y}_t|\mathbf{x}_t, \mathbf{s}_t, \theta_t) \tag{54}$$

- **Parameters of the Model:** For all states s in all source streams,

$$\theta = \{\mathbf{H},\ \mathbf{\Lambda},\ \pi^{(m)},\ \xi^{(m)},\ \mu_s^{(m)},\ \nu_s^{(m)}\} \tag{55}$$

$$\pi^{(m)} = P(S_1^{(m)}) \tag{56}$$

$$\xi^{(m)} = P(S_t^{(m)}|S_{t-1}^{(m)}) \tag{57}$$

There are $K^{(m)}$ states in each stream. The state occupation at time t is presented by a $K^{(m)}$ dimensional vector $S_t^{(m)}$ with 1 in the occupied state and 0 elsewhere. The state transition probabilities from time $t-1$ to t are presented by a $K^{(m)} \times K^{(m)}$ matrix $\xi^{(m)}$. It can be shown that, given the complete dataset, the parameter estimation of the model is tractable.

- **Complete Dataset:** Estimating the complete dataset consists of estimating the statistics associated with the posterior $P(\{\mathbf{x}_t, \mathbf{s}_t\}|\{\mathbf{y}_t\})$ during the E-step of the EM algorithm.

Noiseless Square Invertible Mixing:
In this case $\mathbf{H}$ is square and invertible and $\mathbf{G} = \mathbf{H}^{-1}$ exists. The $D(= M)$ sensor signals are generated by exactly D Gaussian sources which allows the exact factorization

$$\mathbf{x}_t = \mathbf{G}\mathbf{y}_t \tag{58}$$

$$x_t^{(m)} = \sum_{d=1}^{D} G_d^{(m)} y_{t,d} \tag{59}$$

The state occupation probabilities $\gamma_t^{(m)}(s)$ can be estimated by the usual HMM forward-backward procedure in terms of the data $x_t^{(m)}$ from the above estimation. This estimation is therefore tractable.

If one should like to model the D variable time series in terms of fewer M factors, only the largest M principle components can be used. If indeed the sensor signals are generated by only M sources the other components should vanish, or be negligible. One could force this process by applying a non-square $M \times D$ unmixing matrix $\mathbf{G} = \mathbf{H}^{'}(\mathbf{H}\mathbf{H}^{'})^{-1}$.

Mixing with Noise:
When there is mixing noise the unmixing of (58) is no longer possible, since parameters of both $\mathbf{x}$ and $\mathbf{u}$ are unknown in the expression below.

$$\mathbf{x}_t = \mathbf{H}^{-1}(\mathbf{y}_t - \mathbf{u}_t) \tag{60}$$

However, when the mixing is square and invertible with isotropic noise having covariance $\lambda\mathbf{I}$, the problem can be reduced to a tractable estimation with some pre-processing of the data.

In all other cases it can be shown that the posterior estimation is intractable. This is because the estimation requires computing the posterior distributions jointly not only over the source states, but also over the source signals. The exact posterior estimation over the source signals requires summing over all possible source configurations. The intractability stems from the fact that, while the sources are marginally independent, the sources conditional on the observed sensor data are not. Hence, approximation methods such as Variational approximations or Sampling techniques must be used for the posterior estimation.

4.2 BSS Model with Gaussian Sources: Linear Dynamical Systems

Consider the model when the sources are modelled as (single) Gaussian.

The number of sates modelling each source reduces to one. The model reduces to a Linear Dynamical System, where the state space evolution process can be modelled by a single $D \times M$ linear transform $\mathbf{A}$ and the observation generation from the sources can be modelled according to a linear transform $\mathbf{C}$. There are additive stochastic noise processes effecting the state evolution ($\mathbf{w}$) and the observation generation ($\mathbf{v}$) processes. ($\mathbf{C}$ is equivalent to the mixing matrix $\mathbf{H}$ of the previous non-Gaussian source model; the notation is changed to correspond to LDS literature.) The sensor/observation model is now a mixture of M Gaussian's.

$$\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{w}_t \tag{61}$$

$$\mathbf{y}_t = \mathbf{C}\mathbf{x}_t + \mathbf{v}_t \tag{62}$$

4.2.1 Exact Inference is Tractable for the LDS

Having single Gaussian sources makes exact inference tractable for the Linear Dynamical System. Continuous Gaussian hidden state posterior inference is carried out using Kalman Smoothing, a process equivalent to the forward-backward algorithm for inference in the case of the HMM model with discrete hidden states.
Where the hidden states are modelled as a continuous variables, Filtering is computing the posterior probability distribution at time t given all the past observations up to and including time t, $P(\mathbf{x}_t|\{\mathbf{y}_1, \ldots, \mathbf{y}_t\}$, and Smoothing is computing the posterior probability distribution at time t given the entire sequence of observations, $P(\mathbf{x}_t|\{\mathbf{y}_1, \ldots, \mathbf{y}_t, \ldots, \mathbf{y}_T\}$.

For a LDS inference consists of computing the M dimensional mean and the $M \times M$ covariance of the sources. This is done in two steps:

1. Forward recursion: Kalman Filtering to compute $\mathbf{x}_t$ using observations $\{\mathbf{y}_1, \ldots, \mathbf{y}_t\}$.
2. Backward recursion: computing $\mathbf{x}_t$ using observations $\{\mathbf{y}_{t+1}, \ldots, \mathbf{y}_T\}$.

The two posteriors from the two steps are combined to perform Kalman Smoothing, to estimate the posterior of $\mathbf{x}_t$ given the whole sequence of observations $\{\mathbf{y}_1, \ldots, \mathbf{y}_T\}$.

The Kalman Smoother implements the Bayes Rule:

- Given the prior $p(x)$
- and conditional observation $p(y|x)$,
- the Bayes Rule gives the posterior:

$$p(x|y) = \frac{p(y|x)\ p(x)}{p(y)} \tag{63}$$

$$p(y) = Z = \int_x p(y|x)\ p(x)\ dx \tag{64}$$

- Hence Bayes Rule is, to obtain the posterior, multiply the prior by the conditional data likelihood, and re-normalize.

Hence, to implement a Kalman Filter for the model:

- Start with a Gaussian belief on the current state $\mathcal{N}(\mathbf{x}_{t-1}, \mathcal{V}_{t-1})$
- Use the dynamics (61) to convert to a prior over the next state $\mathcal{N}(\mathbf{x}_t^+, \mathcal{V}_t^+)$

- Then condition on the observation to convert this prior into a posterior on the next state $\mathcal{N}(\mathbf{x}_t,\mathcal{V}_t)$

$$
\begin{aligned}
p(\mathbf{x}_t) &= \mathcal{N}(\mathbf{x}_t^+, \mathcal{V}_t^+) && (65)\\
&\quad \mathbf{x}^+ = \mathbf{A}\mathbf{x}_{t-1} && (66)\\
&\quad \mathcal{V}^+ = \mathbf{A}\mathcal{V}_{t-1}\mathbf{A}^{'} \ + \ \mathcal{Q} && (67)\\
&\quad \mathbf{w} \sim \mathcal{N}(0, \mathcal{Q}) && (68)
\end{aligned}
$$

$$
\begin{aligned}
p(\mathbf{y}_t|\mathbf{x}_t) &= \mathcal{N}(\mathbf{C}\mathbf{x}_t, \mathcal{R}) && (69)\\
&\quad \mathbf{v} \sim \mathcal{N}(0, \mathcal{R}) && (70)
\end{aligned}
$$

$$
\begin{aligned}
p(\mathbf{x}_t|\mathbf{y}_t) &= \mathcal{N}(\mathbf{x}_t, \mathcal{V}_t) && (71)\\
\mathbf{x}_t &= \mathbf{x}^+ + \mathbf{K}\,(\mathbf{y}_t - \mathbf{C}\mathbf{x}^+) && (72)\\
\mathcal{V}_t &= (\mathbf{I} - \mathbf{K}\,\mathbf{C})\mathcal{V}^+ && (73)\\
\mathbf{K} &= \mathcal{V}\mathbf{C}^{'}(\mathbf{C}\mathcal{V}_{t-1}\mathbf{C}^{'} \ + \ \mathcal{R})^{-1} && (74)
\end{aligned}
$$

Hence, modelling the sources as single Gaussian renders the computation of the posteriors of the model (inference) tractable for any number of sources.

4.3 Advantages and Disadvantages of Modelling Speech with the Gaussian Source FHMM Model

[60] A number of advantages and disadvantages exist for using the FHMM-BSS model for speech recognition.

1. There is a clear and natural interpretation of the model in terms of mutually uncorrelated/independent source signals and channel conditions generating the sensor signals in the time domain.

2. The models are better models of speech under adverse environmental conditions, in comparison to first order HMM's. They can be trained, in noisy conditions, for better performance in frame classification/segmentation, speech clustering, and segment or phone modelling for speech recognition.

3. Successfully separated sources can then be used for more accurate speech recognition and to generate information on the acoustic environment.

4. We may encounter data sparsity problems in training these models with a high number of parameters. Hence possibilities for parameter tying must be explored.

5. Each of the D sensor signals of the model are assumed to be a sampling of a signal, in the time domain, from a separate microphone recording at the same time instance. In this situation the model has a clear interpretation as each sensor signal being a sum of the source signals at the same time instance, each convolved with the impulse response of the channel to the respective microphone, with added mixing noise.

   However, the speech databases generally used for speech recognition are single microphone recordings. The D features in the time domain can be generated by sampling the signal energy in a short windowed section of the signal from the single microphone every 5 milliseconds or so. Such short segments of speech are generally periodic.

The model interpretation of this situation is as follows. Within the small time frame considered all D sensor signal samples are generated by the same signal samples from M uncorrelated/independent sources, the only variations being due to the impulse responses of the channels or filters. The channel or filter impulse responses, and the uncorrelated/independent source signals are recovered; only a single source sample is recovered for each set of D samples of the sensor signal. This interpretation assumes that changes in the observation signal in the short term is due only to the variations in the impulse response of the channel or filter, and that there is no variation in the sources in the short term. It assumes that the impulse response of the channels/filters change in the short term and only in the short term; the long term variations are due to variations in the source signals. Variation in the sources in the short term can be increasingly accurately modelled by decreasing the time difference between successive windows in the sampling process - i.e. with highly rapid sampling.

6. If speech is sampled in the time domain, sampling must be done at a very high rate (micro-second sampling). Especially for the model interpretation described above, the time interval between windows must be very small, as only one source signal sample is recovered for each window.

   A possible solution would be to adapt the model to a different domain. Consider the cepstral domain in which speech is generally modelled and sampled at a lower rate (milli-second sampling).

7. Speech signals are generally not Gaussian in the time domain. The sources in our model are Mixture of Gaussian models, which can approximate an arbitrary distribution with increasing accuracy with an increasing number of components (i.e. an increasing number of states per stream). However, other options more attractive than increasing the number of source states (with all the accompanying problems of increased time and resources for training, data sparsity problems, overfitting problems with increased number of parameters etc.) can be explored.

   - Preprocess the data by applying an appropriate non-linear transformation first to make the speech distributions more Gaussian in the time domain.
   - Model the data in a domain in which speech distributions are Gaussian, such as in the cepstral domain.

8. The model assumes that the sensor signals (observations) are linearly related to the sources. As discussed, this is accurate in the time domain. However, this assumption may be inaccurate in a new domain in which the model operates. A solution to this is to develop a Non-linear Dynamical Model for speech in the domain BSS is non-linear.

4.4 Model Adaptation to the Cepstral Domain

[60]A digital speech signal vector of length n, $x[n]$ is obtained by windowing the speech signal every 5 ms or so with a window of length N (about 20-30 ms), and transforming into the frequency domain using the short-time Fourier analysis. For a periodic signal in the time domain, the Fourier transform is a discrete periodic signal in the frequency domain. The transform is taken using a filterbank with N filters.

$$X_a[k] = \sum_{n=0}^{N-1} x[n]e^{-j2\pi nk/N} \quad , 0 \leq k \leq N \tag{75}$$

The inverse Fourier transform is taken of the log of the digital signal in the frequency domain $\ln|X[k]|$, to give the real cepstrum $c[n]$ of the digital speech signal $x[n]$.

$$c[n] = \frac{1}{N}\sum_{n=0}^{N-1} \ln|X_a[k]|e^{j2\pi nk/N} \quad , 0 \leq n \leq N \tag{76}$$

The Mel Frquency Cepstrum (MFCC) is a speech representation defined similar to the real cepstrum above, the difference being that the D filter of the filterbank have a nonlinear frequency scale to approximate the behavior of the auditory system. Such filters compute the average spectrum around each k^{th} center frequency with increasing bandwidths. The MFC cepstrum is computed using the log-energy $S[r]$ at the output of each filter:

$$S[r] = \ln\left[\sum_{k=0}^{N-1} |X_a[k]|^2 H_d[k]\right] \quad , 0 < r < R \tag{77}$$

$$c_{mfc}[n] = \sum_{r=0}^{R-1} S[r] \cos\left(\pi(r-0.5)\frac{n}{R}\right) \quad , 0 < r < R \tag{78}$$

A transform that converts a convolution into an addition is defined as a homomorphic transform. It can be seen that signals that are convolutional in the time domain are additive in the real cepstral domain. The MFCC transform defined using the actual energy (77) is approximately homomorphic.

Model Transformation

Since the relations between the sources change with a homomorphic transformation, the model must be adapted for the real cepstral and mfcc domains.

Consider the time domain signal corrupted with both convolutional and additive noise $y_t[d]$. Let $\mathbf{y}_t, \mathbf{x}_t, \mathbf{h}_t, \mathbf{u}_t$ be corresponding D length time domain vectors.

$$y_t[d] = x_t[d] * h_t[d] + u_t[d] \tag{79}$$

$$\mathbf{y}_t = \mathbf{x}_t * \mathbf{h}_t + \mathbf{u}_t \tag{80}$$

An approximating transformation into the real cepstral domain can be made as follows. Taking the log norm of the short-time Fourier transforms,

$$\begin{aligned} |Y[k]| &= |X[k]||H[k]| + |U[k]| \\ \ln|Y[k]| &= \ln|X[k]| + \ln|H[k]| + \ln\left(1 + exp\left(\ln|U[k]| - \ln|X[k]| - \ln|H[k]|\right)\right) \end{aligned} \tag{81}$$

For the (homomorphic) MFC cepstral domain, taking the log of the square norm, where θ is the angle between the real signal and the noise, and noting that when they are statistically independent $\theta = \pi/2$:

$$\begin{aligned} |Y[k]|^2 &= |X[k]|^2|H[k]|^2 + |U[k]|^2 + 2|X[k]||H[k]||U[k]|cos(\theta_k) \\ &= |X[k]|^2|H[k]|^2 + |U[k]|^2 \end{aligned}$$

$$\begin{aligned} \ln|Y[k]|^2 &= \ln|X[k]|^2 + \ln|H[k]|^2 \\ &\quad + \ln\left(1 + exp\left(\ln|U[k]|^2 - \ln|X[k]|^2 - \ln|H[k]|^2\right)\right) \end{aligned}$$

Let $\mathbf{y}_c, \mathbf{x}_c, \mathbf{d}_c, \mathbf{u}_c$ be N length cepstrum vectors defined similar to $\mathbf{x}_c$ below, where $\mathbf{C}$ is a $N \times M$ transform where M is the length of the frequency domain vector.

$$\mathbf{x}_c = \mathbf{C}\left(\ln|X(f_0)|^2, \ldots, \ln|X(f_{M-1})|^2\right)' \tag{82}$$

$$\text{where } \mathbf{C} = [c_{ij}] = \left[cos\left(i(j-0.5)\frac{\pi}{M}\right)\right] \tag{83}$$

Hence, the time domain expression of (80) transforms to the cepstral domain expression:

$$\mathbf{y}_c = \mathbf{x}_c + \mathbf{h}_c + g(-\mathbf{x}_c - \mathbf{h}_c + \mathbf{u}_c) \tag{84}$$

$$\text{where } g(\mathbf{z}) = \mathbf{C}\ \ln\left(1 + e^{\mathbf{C}^{-1}\mathbf{z}}\right) \tag{85}$$

Consider the mixing expression (43) for the FHMM BSS probability model:

$$y_{t,d} = \sum_{m=1}^{M} h_d^{(m)} * x_t^{(m)} + u_{t,d} \tag{86}$$

However, according to our model, only a single source signal is recovered per a D dimensional sensor signal vector. Let the recovered signal be the mean of an N length cepstrum vector $\mathbf{x}_c^{(m)}$ defined as in (82).

$$x_c^{(m)} = \frac{1}{N} \sum_{n=0}^{N-1} \mathbf{x}_c^{(m)}[n] \tag{87}$$

The linear expression (86) in the time domain transforms into the non-linear expression in the cepstral domain according to the following recursive algorithm:

$$\begin{aligned}
y_{c,d} &= y_{c,d}^{(1)} \\
y_{c,d}^{(M)} &= x_c^{(M)} + h_d^{(M)} + g\left(-x_c^{(M)} - h_d^{(M)} + u_{c,d}\right) \\
y_{c,d}^{(m)} &= x_c^{(m)} + h_d^{(m)} + g\left(-x_c^{(m)} - h_d^{(m)} + y_{c,d}^{(m+1)}\right) \quad \text{, for } m = (M-1)\ldots 1 \\
g(z) &= \ln(1 + e^z)
\end{aligned}$$

In the above algorithm, the factor indexed by 1 corresponds to the source that has the highest weighted principal component and factor weighted M corresponds to the lowest weighted component; there is decreasing weight on principal component with increasing index m.

Since the non-linear function g is invertible, the sources can be recovered in the cepstral domain by directly reversing the recursive algorithm *only* in the situation where there is no mixing noise. The non-linear function g can be approximated by using Taylor series expansion.

The recursive algorithm above is a non-linear function f coupling the sources generating the observations. This non-linear function f can be estimated by Taylor series expansion to estimate g and, in turn, by direct recursion to compute f.

Hence, we have shown that the Blind Source Separation problem is indeed non-linear in the cepstral domain and is more accurately modelled by a FHMM Model with non-linear mixing. In this section we have shown that the BSS model is non-linear in the cepstral domain by arguing from the principles of signal processing. We have derived the exact non-linear function $f()$ coupling the sources to generate the observed signal (in the case of noiseless square invertible mixing), and (will) have shown that f can be approximated by the Taylor series expansion of g.

Indeed, a general Non-Linear Dynamical State Space modelling framework exists for modelling such processes, where the non-linearities can be learned as part of an Expectation Maximization algorithm. These models have the added advantage that a mixing noise model can be learned as part of the learning algorithm. We shall look at such models in the next section.

However, a linear function is an approximation, by regression, of a non-linear function. For the reason that we seek the simplest model in accordance with Occam's Razor it is indeed very possible, even likely, that a BSS model with a linear generation of the sensor signal generalizes better than a true model with a non-linear generation of the sensor signal. Hence, two models - a linear dynamical model (with a single Gaussian source model) and a model with a Mixture of Gaussian source model are studied, both with a linear generative model of the sensor signal in the cepstral domain.

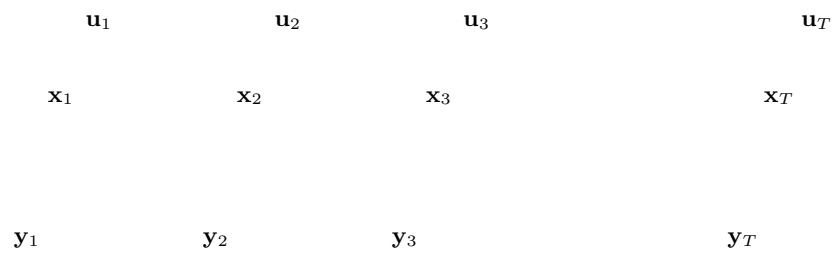


Figure 9: A dynamical state space model with hidden sources $\mathbf{x}_t$, inputs $\mathbf{u}_t$ and observations $\mathbf{y}_t$.

5 Nonlinear Dynamical State Space Models for Speech

[30] [31] As shown in the previous section, the dynamics underlying the processes generating a sequence of observations is not always correctly modelled by a linear model. A Nonlinear Dynamical State Space Model firstly allows the process that generates the observation signal at a time instance to be non-linear. Secondly, it allows the underlying dynamics to evolve in time in a non-linear manner.

Let vector $\mathbf{x}_t$ represent the M dimensional source state space with M streams and let $\mathbf{y}_t$ be the D dimensional observation vector. To allow for more generality, let $\mathbf{u}_t$ be an external input at time t affecting the generation of the observation at time t as well as the dynamics of evolution between time instances t and $t+1$. Let $\mathbf{v}_t$ and $\mathbf{w}_t$ be noise processes contributing to the observation generation and the dynamics of the evolution respectively.

$$\mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t) + \mathbf{w}_t \tag{88}$$

$$\mathbf{y}_t = g(\mathbf{x}_t, \mathbf{u}_t) + \mathbf{v}_t \tag{89}$$

This is a comprehensive input driven, stochastic (noisy) generative model for time series data. For a true generative model, stochasticity is essential to allow a model with a few parameters to generate a rich variety of outputs. The M internal (hidden) source states $\mathbf{x}_t$ capture the deterministic portion of the underlying processes generating the observation at time t. The noise portion $\mathbf{v}_t$ models the non-deterministic, chaotic processes influencing the generation of the observation. Finally the external inputs $\mathbf{u}_t$ allow an external process to influence the generation of the observation. Similarly deterministic, chaotic and externally input factors are allowed to affect the dynamics of evolution of the underlying hidden processes through time. For speech modelling we shall disregard the influence through external processes by setting $\mathbf{u}_t$ to zero vectors.

In a non-linear dynamical system, both f and g are allowed to be non-linear functions. These vector valued non-linear functions are assumed to be differentiable, but otherwise arbitrary. They are time invariant; i.e. f and g are constant through time.

Any non-linear function can be modelled to arbitrary accuracy as a succession of locally linear functions. These locally linear functions can be modelled by Radial Basis functions, and the non-linear function can be fitted with a regression of RBF functions by an RBF network. This approximation is termed local linearization (Extended Kalman Smoothing). Gaussian Radial Basis Function approximators are commonly used for the regression.

Learning the Nonlinear Dynamical Model is carried out using an Expectation Maximization algorithm. In the Expectation step the algorithm estimates the state occupation probabilities given the non-linear functions f, g and the parameters of the model θ. In the maximization step it fits f and g with RBF networks by regressions, as well as re-estimating the parameters θ, both in order to maximize the likelihood of the observed data. Of course, the expectation step is intractable and

must use an approximating technique such as variational approximation or stochastic sampling. The nonlinear maximization step also becomes intractable because it requires integrating out the uncertainty in the hidden states. However, by using Gaussian Radial Basis Function approximators to model the nonlinearities the integral becomes tractable, allowing the maximization to be solved via a system of linear equations.

Instead of deterministically estimating the non-linear mixing function f as in the BSS model in the previous section, the non-linearity f is *learned* by regression as an RBF network as part of the EM process.

While in the FHMM BSS model the internal dynamics for state space evolution through time is modelled as linear (with the $\xi^{(m)}$ parameters), it is possible to allow these dynamics to also be modelled as a non-linear function g that can be fitted by an RBF network during the Expectation Maximization process.

6 Software

6.1 Graphical Models Toolkit:

An open source software system using Graphical Models for speech and time-series processing by Jeff Bilmes (University of Washington) and Geoff Zweig (IBM). [61] [63]

- The GMTK software specifies a generalized Dynamic Bayesian Network (DBN) modelling framework for Automatic Speech Recognition. The software has been extensively used in 2001 and 2002 Johns-Hopkins Summer Workshops for sophisticated speech recognition research. The SPINE ASR system has been built using the software. [62]
- Discrete hidden states, having 1 - 4000 possible discrete values, are implemented for hidden state models. Discrete, continuous Gaussian and continuous Mixture of Gaussian observation models are implemented.
- Implements a Generalized Expectation Maximization algorithm for inference and parameter estimation which is defined as being economical in the use of both time and memory resources for exact inference. The algorithm does exact inference in logarithmic space; beam pruning is possible. This algorithm can efficiently implement 2 stream FHMM model learning.
- Mixture model observation distributions are trained with a splitting-vanishing algorithm.
- Streams are supported.
- Parameter tying is allowed at all levels.
- Gibbs sampling is implemented and described as inexpensive. Hence, sampling is used as the intractable model training technique. Other sampling techniques, if necessary, can be easily implemented by modifying this module.
- The main implementation requirement would be to implement Variational Approximation techniques for intractable inference. This would be for models with more than 2 streams.
- The facilities for modelling non-linearity by using switching parent state variables are in place.
- Regularization can be carried out with ease by modifying the EM algorithm, by adding parameter based penalization factors to the objective function.
- A Viterbi decoder has been implemented and can be used for decoding in both Segmentation and Speech Recognition experiments.
- The software is closely modelled after HTK and accepts HTK data files as input/output files.

6.2 HTK

[65] There is a version of HTK implemented for linear cepstral domain frame modelling using Factor Analysis by Mark Gales and Anti Rosti. [16] The mixing and unmixing in this model is linear. This implementation is modified with a non-linear unmixing and mixing module to implement the FHMM BSS Model.

		#Cl.	**spkr**	**spkr-env2**	**spkr-env3**	**spkr-music**
	#Labels		77	103	117	116
adapt_c:	η_{BBN} $[0,1]$	81	0.732	0.684	0.675	0.681
	η_{BBN} $[-1,1]$		0.514	0.390	0.353	0.365
speaker1_c:	η_{BBN} $[0,1]$	92	**0.760**	0.715	0.699	0.702
	η_{BBN} $[-1,1]$		**0.588**	0.468	0.418	0.425
speaker2_c:	η_{BBN} $[0,1]$	165	0.668	0.628	0.613	0.623
	η_{BBN} $[-1,1]$		0.569	0.461	0.417	0.436

Table 2: BBN [0,1] and [-1,1] Scaled Efficiency Results of Speaker-Environment Clustering

7 Experiments

7.1 FHMM Models for Speaker-Environment Clustering

The first experiments are conducted in clustering with respect to speech and the environment to judge the effects of the acoustic environment in clustering.

The data clustered are 488 segments of 1996 Hub-4 Broadcast News Transcription development (BNdev97) data homogeneous with respect to the speaker and the environment. A speech representation of 39 PLP features is used with log energy , delta coefficients and acceleration coefficients. The clustering unit is the segment.

The model built is a single Gaussian model (hence a first order or single source model) with a mean and a covariance matrix for each segment. The segments are homogeneous with respect to the speaker and the environment. A Gaussian model for each cluster is built using all of the speech frames in each cluster.

Varied clustering schemes are used for clustering with the aim of producing perfect speaker clusters. The Arithmetic Harmonic Sphericity measure is used to measure the distance between two Gaussian's. The segments are pre-classified into 4 categories according to gender and bandwidth and the 3 clustering schemes are applied to each category. A top-down splitting and merging schedule is then used to produce clusters with high speaker purity. Three clustering schemes are used to produce clusters with the highest possible speaker purity. The 'adapt_c' scheme clusters by growing a clustering tree that terminates on a minimum occupancy count. The 'speaker1_c' scheme grows a tree until the gain from growth falls below a threshold, then recombines nodes that have a distance between the nodes that is less than twice the average distance within the node. The 'speaker2_c' scheme grows a larger tree that terminates on a larger gain growth threshold, and allows fewer recombination of nodes that have an inter-node distance less than the average within-node distance.

The produced clusters are then assessed according to the speaker and the environment to evaluate the effect of the acoustic environment on the clusters. The metric used is the BBN efficiency metric carried out at the frame level. The clustering scheme which grows a tree of moderate depth with re-combinations, while achieving a speaker level clustering with the highest BBN ([0,1] scaled) efficiency of 0.760, also tracks the speakers through a noisy/clear environment with a relatively high BBN efficiency of 0.715. The study of the clusters confirmed there is a secondary level clustering being achieved according to the noise condition; i.e. clusters with the same speaker are split into several clusters with noise and without noise. Merging them together by variations in splitting or recombining parameters is difficult.

The cause for this sub-optimal clustering is that the single stream Gaussian model is an inadequate model for both speech and the acoustic environment.

Consider, for instance, training a 2 stream model with a separate Gaussian model for speech and a second Gaussian model for the environment for each segment. Clustering by taking the distance measures between the Gaussian's for speech would reduce the effects of the environment

	Env2	Env3	Music
Correctly Classified Segments	88.6%	81.9%	79.7%
Correctly Classified Frames	90.5%	84.4%	82.8%

Table 3: Recognition results of environmental classification for the 3 environmental schemes.

Segment Confusion	Classified Clear	Classified Noise
Labelled Clear	349	46
Labelled Noise	17	141
Frame Confusion	Classified Clear	Classified Noise
Labelled Clear	414364	35439
Labelled Noise	22893	141814

Table 4: Recognition confusion matrices for env2 labelling scheme

and give purer and more efficient speaker clusters. In-turn, clustering using distance measures between the Gaussian's representing environmental noise would give purer clusters of the noise conditions. The initial experiments have confirmed this.

This would greatly increase the accuracy of speech clustering and should in turn increase the performance of speech recognition systems adapted to speakers, as clustering is used extensively for speaker adaptation. This also provides a way of accurately clustering according to the environment for more accurate environment adaptation of speech recognition systems, in turn increasing the accuracy of speech recognition in varied noise environments.

7.2 FHMM for Acoustic Environment Identification

In the previous clustering experiments the source model is a single stream Gaussian.

Not always can the speech or the environment signal be represented by a single Gaussian model. However, a Gaussian Mixture Model can model any signal to arbitrary accuracy. In this section we experiment with the ability to learn the environment signal with a GMM.

ML Gaussian Mixture Model classifiers were trained on 1997 US Broadcast New training data, to identify the background conditions of the 553 automatic segments of 1997 US Broadcast News development data. The segments have speech in the foreground.
Environment Schemes:

- **Env2**: Noise/ Clear
- **Env3**: High Noise/ Low Noise/ Clear
- **Music**: Music/ Other Noise/ Clear

The classification results are reported in 3. It is observed on further investigation of the misclassifications (as shown in the confusion matrices in 4 through 6) that the foreground speech signal interfered with the identification of the background noise condition. For example Clear segments misclassified as Noise belonged to one speaker with a distinctly "hoarse" voice.

It is clear that this interference effects of the speech signal can be reduced by training (for instance) a 2 stream FHMM model, where each stream is learned as a GMM.

Hence, experiments in Clustering first order (single stream) models of speech for Speaker-Environment turns and the use of a first-order GMM for background environment identification have shown evidence that:

- The background noise condition shows a significant effect on clustering first order models of speech

Segment Confusion	Classified Clear	Classified High Noise	Classified Low Noise
Labelled Clear	349	10	36
Labelled High Noise	2	23	15
Labelled Low Noise	15	22	81
Frame Confusion	Classified Clear	Classified High Noise	Classified Low Noise
Labelled Clear	414364	4707	30732
Labelled High Noise	2880	14043	15667
Labelled Low Noise	20013	21351	90753

Table 5: Recognition confusion matrices for env3 labelling scheme

Segment Confusion	Classified Clear	Classified Music	Classified Other Noise
Labelled Clear	339	11	45
Labelled Music	2	24	34
Labelled Other Noise	12	8	78
Frame Confusion	Classified Clear	Classified Music	Classified Other Noise
Labelled Clear	400334	13192	36277
Labelled Music	3853	22115	27919
Labelled Other Noise	15021	9547	86252

Table 6: Recognition confusion matrices for music labelling scheme

- The foreground speech signal interferes with the identification of the background noise condition when using a first order GMM.

Hence there is evidence that the first order HMM/GMM model is an inadequate model, or too simple a model, of speech in noisy acoustic conditions. The forthcoming experiments will explore the use of higher order HMM's that explicitly model both speech and noise sources with increasing complexity.

7.3 FHMM's for Segmentation & Speech Recognition

In most current segment generators a pre-segmentation task is carried out to perform audio-type classification of the frames. In the Cambridge segmenter this stage classifies the audio stream frames as wideband speech (S), Music(M), music and speech (MS) or telephone (T) speech frames. Following frame labelling as above, the M frames are discarded, and adjacent frames with the same label are grouped together to form segments. The segments are marked as wideband speech (S), telephone speech (T), speech with music in the background (MS) or, from the discarded adjacent frames, pure music (M) segments. The primary goals in audio type classification are:

i. Minimize loss of speech by minimizing misclassification of other frames as music (M) frames.

ii. Break the audio stream into reasonable size segments that are homogeneous with respect to the foreground speech and background environment.

iii. Maximize the purity of the segments by minimizing misclassification of all frames.

The initial segmentation experiments designed using FHMM models improve on the current segmentation described above by using a more accurate speaker-environment based segmentation process.

This is a preliminary system that will be built-up and refined to build a more complex speech recognition system in noisy acoustic conditions. The system will be built up by changing the speech model, which is currently a gender model, to a mono-phone model and, in turn, a tri-phone model.

The experiments have been set up using GMTK software.

The data used is the Broadcast News 97 training/development/eval dataset. The whole of the training dataset is needed for training as there are a large number of states that need to be trained because of the environment being represented.

Initially, a 2 stream speech-environment representation is used. This FHMM is tractable and can be learned without approximation.

Three representations of the environment are considered:
- Env2: 2 states Noise/Clear
- Env3: 3 states High-Noise/Low-Noise/Clear
- Music: 3 states Music/Other-Noise/Clear

Two representation of speech are considered:
- Gender: 2 states for M/F
- Monophone: 47 states for monophones

The observation model is a GMM that are be trained by EM multiple iterations with splitting (of the top highly weighted components) and vanishing (of the bottom low weighted components) until convergence.

The Gender-Env models trained are used for:

(1) Improved Environment identification of Broadcast News segments, to be compared against the results of the section above.

(2) Improved gender-environment based segmentation of the BN data stream, to be compared against the results of the current segmenter

From the initial test runs the experiments show extremely good recognition rates of the Gender of each utterance. This is likely due to the separation of the Environment effects which interfere with the recognition of Speech. This results in an increase in the segmentation accuracy of the current gender based segmenter.

The recognition results on the Environment is less accurate. This is likely because the environment signal is weak and much more data is necessary for the training. The segmentation accuracy is further increased by using gender-environment based segmentation.

Unlike a single stream GMM Speech-Environment model, there are two separate transition matrices controlling the switching of the Environment state and Speech state in the new model. These matrices can separately and individually tuned for improved performance.

Since the model infers a separate environment model, it generalizes better to unseen Environment-Speech state combinations. This is shown by an improvement over a single stream GMM trained with a state for each Environment-Speech combination.

References

[1] Nock H.J. (2001): *Techniques for Modelling Phonological Processes in Automatic Speech Recognition* Phd thesis, Cambridge University Engineering Department.

[2] Nock H.J. and Young S.J. (2000): Loosely Coupled Hidden Markov Models for ASR *Proc. ICSLP 2000,* Beijing, China.

[3] Ghahramani Z. (1995): Factorial Learning and the EM ALgorithm. In Tesauro G., Touretzky D.S. and Alspector J. (Eds.), *Advances in Neural Information Processing Systems 7,* pp. 617-624. Margan Kaufmann, San Francisco, CA.

[4] Ghahramani Z. and Jordan M.I. (1996): Factorial Hidden Markov Models. In Touretzky D.S., Mozer M.C. and Hasselmo M.E. (eds.), *Advances in Neural Information Processing Systems 8,* MIT Press, Cambridge MA

[5] Heckerman D. (1995): A Tutorial on Learning With Bayesian Networks. *Technical Report MSA-TR-95-06,* Microsoft Research. Availabel from http://research.microsoft.com/ heckerman/

[6] Jaakkola T.S. and Jordan M.I. (2000): Bayesian Parameter Estimation via Variational Methods. *Statistics and Computing 10,* pp. 25-37.

[7] Jaakkola T.S. and Jordan M.I. (1999): Improving the Mean Field Approximation via the use of Mixture Distributions. In M. I. Jordan (Ed.), *Learning in Graphical Models.* Cambridge: MIT Press. Available from http://www.cs.berkeley.edu/ jordan/publications.html

[8] Jordan M.I. (1998): AAAI Tutorial on Graphical Models and Variational Approximation. Department of Electrical Engineering and Computer Science, University of California, Berkeley. http://www.cs.berkeley.edu/ jordan/publications.html.

[9] Jordan M.I., Ghahramani Z., Jaakkola T.S. and Saul L.K. (1999): An Introduction to Variational Methods for Graphical Models. In M. I. Jordan (Ed.), *Learning in Graphical Models.* Cambridge: MIT Press. Available from http://www.cs.berkeley.edu/ jordan/publications.html

[10] Saul L.K. and Jordan M.I. (1997): A Variational Principle for Model-based Interpolation. In Mozer M.C., Jordan M.I. and Petsche T. (Eds.), *Advances in Neural Information Processing Systems 9.* MIT Press, Cambridge MA

[11] Saul L.K. and Jordan M.I. (1996): Exploiting Tractable Sunstructures in Intractable Networks. In Touretzky D.S., Mozer M.C. and Hasselmo M.E. (Eds.), *Advances in Neural Information Processing Systems 8,* MIT Press, Cambridge MA

[12] Smyth P., Heckerman D. and Jordan. M.I. (1997): Probabilistic Independence Networks for Hidden Markov Probability Models. *Neural Computation 9,* pp. 227-270.

[13] Bilmes J. (2000): Dynamic Bayesian Multi-Nets. *Proc. The 16th Conference on Uncertainty in Artificial Intelligence,* Stanford, CA.

[14] Bilmes J. (1999): *Natural Statistical Models for Automatic Speech Recognition.* Ph.D. Thesis, Dept. of EECS, CS Division, U.C. Berkeley. Avaliable from http://ssli.ee.washington.edu/people/bilmes/pubs-frame.html.

[15] Logan B.T. and Moreno P.J. (1998): Factorial HMMs for Acoustic Modellling. *Proc. ICASSP,* pp. 813-816.

[16] Rosti A-V.I. and Gales M.J.F. (2002): Factor Analysed HMMs. *Proc. ICASSP 2002.*

[17] Zweig G.G. (1998): *Speech Recognition with Dynamic Bayesian Networks.* Phd thesis, UC Berkeley Engineering catalogue no. T7.6.1998.Z945. Available from http://www.cs.berkeley.edu/ zweig/.

[18] Zweig G.G. and Russell S. (1999): Probabilistic modeling with Bayesian networks for automatic speech recognition. *Australian Journal of Intelligent Information Processing Systems,* 5(4), 253-60, 1999 (invited paper).

[19] Attias H. (1999): Independent Factor Analysis. *Neural Computation 11(4),* pp. 803-851.

[20] Attias H and Schreiner C.E. (1998): Blind Source Separation and Deconvolution: the Dynamic Component Analysis Algorithm. *Neural Computation 10,* pp. 1373-1424

[21] Attias H. (2000): Independent Factor Analysis with Temporally Structured Factors. *Advances in Neural Information Processing Systems 12* (Ed. by T. Leen et al.). MIT Press, Cambridge, MA.

[22] Attias H., Platt J.C., Acero A. and Li Deng (2001): Speech Denoising and Dereverberation using Probabilistic Models. *Advances in Neural Information Processing Systems* 13 (Ed. by T. Leen). MIT Press, Cambridge, MA

[23] Attias H., Li Deng, Acero A. and Platt J.C. (2001): A New Method for Speech Denoising and Robust Speech Recognition using Probabilistic Models for Clean Speech and for Noise. em Proc. Eurospeech 2001

[24] Pearlmutter B.A. and Parra L.C. (1996): Maximum Likelihood Blind Source Separation: A Context-Sensitive Generalization of ICA. *International Conference on Neural Information Processing,* Springer-Verlag, Hong Kong.

[25] Roweis S. and Ghahramani Z. (1999): A Unifying Review of Linear Gaussian Models. *Neural Computation 11,* pp. 305-345.

[26] Rubin D.B. and Thayer D.T. (1982): EM Algorithms for ML Factor Analysis. *Psychometrika 47(1),* pp. 69-76.

[27] Tipping M.E. and Bishop C.M. (1997): Probabilistic Principal Component Analysis. *Technical Report NCRG/97/10,* Microsoft Research. http://research.microsoft.com

[28] Frey B.J. and Hinton G.E. (1999): Variational Learning in Nonlinear Gaussian Belief Networks. *Neural Computation 11:1,* pp. 193-214.

[29] Giannakopoulos X. and Valpola H. (2001): Nonlinear Dynamical Factor Analysis. In *Proc. of The Twentieth International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering,* MaxEnt 2000, Paris, France, pp. 305-317.

[30] Ghahramani Z. and Roweis, S. (2000): An EM Algorithm for Identification of Nonlinear Dynamical Systems. To appear in *Kalman Filtering and Neural Networks.* Simon Haykin.

[31] Ghahramani Z. and Roweis, S. (1999) Learning Nonlinear Dynamical Systems Using an EM Algorithm. In M. S. Kearns, S. A. Solla, D. A. Cohn, (eds.) *Advances in Neural Information Processing Systems 11:* pp. 599-605. MIT Press.

[32] Ghahramani Z. and Hinton G.E. (1998): Hierarchical Nonlinear Factor Analysis and Topographic Maps. In Jordan M.I, Kearns M.J. and Solla S.A. (eds.) *Advances in Neural Information Processing Systems 10.* MIT Press: Cambridge, MA.

[33] Ghahramani Z. and Hinton G.E. (1996): Switching State Space Models. *Tech. Report CRG-TR-96-3,* Dept. of Computer Science, University of Toronto, July 1996.

[34] Hinton G.E. and Ghahramani Z. (1997): Generative Models for Discovering Sparse Distributed Representations. *Philosophical Transactions Royal Society B, 352:* pp. 1177-1190.

[35] Murphy K. (1998): Switching Kalman Filters. *Tech. Report.* Department of Computer Science, University of California, Berkeley, August 1998.

[36] Roweis S. (1999): Constrained Hidden Markov Models. *Proc. Neural Information Processing Systems 12 (NIPS'99),* pp. 782-788.

[37] Attias H. (1999): Inferring Parameters and Structure of Latent Variable Models by Variational Bayes. *Proc. 15^{th} Conference on Uncertainty in Artificial Intelligence.*

[38] Beal M. J. and Ghahramani Z. (2002): The Variational Bayesian EM Algorithm for Incomplete Data: with Application to Scoring Graphical Model Structures. In Bernardo J.M. et. al (eds.) *Bayesian Statistics 7,* Oxford University Press, 2003. To appear.

[39] Everitt B.S. (1984): *An Introduction to Latent Variable Models.* Chapman and Hall, London.

[40] Freidman N., Murphy K. and Russell S. (1998): Learning the Structure of Dynamic Probabilistic Networks. *Proc. Uncertainty in AI, fourteenth conference,* Madison, Wisconsin. Morgan Kaufmann.

[41] MacKay J.D.C (1992): Bayesian Interpolation. *Neural Computation* **4**, pp. 415-447.

[42] Neal R.M. and Hinton G.E. (1998): A View of the EM Algorithm that Justifies Incremental, Sparse and Other Variants. In Jordan M.I.(Ed.), *Learning in Graphical Models*, Kulwer Academic Press.

[43] Woodland P., Gales M.J.F., Wickramaratna J.T., Yuan D., Federico M., Bertoldi N., Brugnara F., Giuliani D., Gauvain J.L., Lamel L., Adda G., Lefevre F., Schwenk H., Ney H. and Pitz M. (2002): *Report on Acoustic Meta-data Markup.* Coretex Project on Improving Core Speech and Recognition Technology. Available from http //coretex.itc.it/public/frames/f_results.htm.

[44] Young S.J., Odell J.J. and Woodland P.C (1994): Tree Based State Tying for High Accuracy Acoustic Modelling. *Proc Human Language Technology Workshop,* Plainsboro NJ, Morgan Kaufman Publishers Inc, 307-312.

[45] Freitas de N., Hjen-Srensen P., Jordan M.I. and Russell S. (2001): Variational MCMC. In Breese J. and Koller D.(Eds.), *Proc. Seventeenth Conference of Uncertainty in Artificial Intelligence.*

[46] Gilks W.R., Richardson S. and Speigelhalte D.J. (1996): *Markov Chain Mote Carlo in Practice.* Chapman and Hall, Suffolk.

[47] Kanazawa K., Koller D. and Russell S. (1995): Stochastic Simulation Algorithms for Dynamic Probabilistic Networks. *Proceedings of the 11th Uncertainty in AI Conference,* Montreal, Canada.

[48] Neal R.M. (1993): Probabilistic Inference Using Markov Chain Monte Carlo Methods, *Technical Report CRG-TR-92-1,* Dept. of Computer Science, University of Toronto.

[49] Neal R.M. (1994): Sampling from Multimodal Distributions Using Tempered Transitions, *Technical Report CRG-TR-93-1,* Dept. of Computer Science, University of Toronto.

[50] Chen C.-P., Filali K. and Bilmes J. (2002): Frontend Post Processing and Backend Model Enhancement on the Aurora 2.0/3.0 databases. *Proc. ICSLP'2002,* Denver Colorado, pp. 241-244.

[51] Evans N.W.D. and Mason J.S. (2002): Computationally Efficient Noise Compensation for Noise Robust Speech Recognition Assessed Under the Aurora 2/3 Framework. *Proc. ICSLP'2002,* Denver Colorado, pp. 485-488.

[52] Gales M.J.F. and Young S.J. (1993): HMM Recognition in Noise Using Parallel Model Combination. *Proc. Eurospeech 1993,* pp. 837-840.

[53] Ida M. and Nakamura S. (2002): HMM Composition Based Rapid Model Adaptation Using A Priori Noise GMM Adaptation: Evaluation on Aurora 2 Corpus. *Proc. ICSLP'2002,* Denver, Colorado, pp. 437-440.

[54] Kim H.K. and Rose R.C. (2002): Evaluation of Robust Speech Recognition Algorithms for Distributed Speech Recognition in a Noisy Automobile Environment. *Proc. ICSLP'2002* Denver, Colorado, pp. 233-236.

[55] Segura J.C., Benftez M.C., de la Torre A. and Rubio A.J. (2002): Feature Extraction Combining Spectral Noise Reduction and Cepstral Histogram Equalization for Noise Robust ASR. *Proc. ICSLP'2002,* Denver, Colorado, pp. 225-228.

[56] Hain T. and Woodland P.C. (1999): Dynamic HMM Selection for Continuous Speech Recognition. *Proc. Eurospeech 1999,* pp. 532-535.

[57] Nock H.J. and Young S.J. (2001): A Comparison of Exact and Approximate Algorithms for Decoding and Training Loosely Coupled HMMs. *Proc. of WISP (Institute of Acoustics) 2001,* Stratford Upon Avon.

[58] Cover T.M. and Thomas J.A. (1991): *Elements of Information Theory.* John, Wiley and Sons, New York.

[59] Harville D.A. (1997): *Matrix Algybra from a Statistician's Perspective.* Springer-Verlag, New York.

[60] Huang X., Acero A. and Hon H.-W. (2001): *Spoken Language Processing: A Guide to Theory Language, and System Development.* 2001 Prentice Hall, New Jersey.

[61] Bilmes J. and Zweig G. (2002): The Graphical Models Toolkit: An Open Source Software System for Speech and Time-Series Processing *Proc. ICASSP 2002,* Orlando, Florida.

[62] Bilmes J. and Zweig G. (2002): The 2001 GMTK-Based SPINE ASR System. *Proc. International Conference on Spoken Language Processing (ICSLP) 2002.* Denver, Colorado.

[63] Bilmes J. and Zweig G. (2002): *The Graphical Models Toolkit, Documentation, and Aurora Tutorial.* Draft release. Avaliable from http://ssli.ee.washington.edu/∼bilmes/gmtk/.

[64] Murphy K.P. (2001): The Bayes Net Toolbox for Matlab. *Compuing Science and Statistics,* vol. 33.

[65] Woodland P.C., Hain T., Johnson S.E., Neisler T.R., Tuerk A., Whittaker E.W.D. and Young S.J.(1998): *The 1997 HTK Broadcast News Transcription System.* Proc. DARPA Broadcast News Transcription and Understanding Workshop pp.41-48 Lansdowne, Virginia

[66] Young S.J., Evermann G., Kershaw D., Odell J., Ollason D., Valtchev V. and Woodland P.C. (2001): *The HTK Book (for HTK Version 3.1).* Entropic Labs and Cambridge University Engineering Department. Avaliable from http://htk.eng.cam.ac.uk/.