Temporally Varying Model Parameters for Large Vocabulary Continuous Speech Recognition

K. C. Sim and M. J. F. Gales

Department of Engineering, University of Cambridge
Trumpington Street, CB2 1PZ Cambridge, England
{kc323,mjfg}@eng.cam.ac.uk

Abstract
Many forms of time varying acoustic models have been investigated for speech recognition. However, there has been little success in applying these models to Large Vocabulary Continuous Speech Recognition (LVCSR). Recently, fMPE was introduced as a discriminative feature space estimation scheme for the HMM-based LVCSR. This method estimates a projection matrix from a high dimensional space (∼100,000) down to a standard feature space (typically 39). This projection is then added on to the original feature vector (e.g. MFCC or PLP) to yield a feature vector to train the final model. This paper considers IMPE as a time varying model for the mean vectors by applying the time varying feature offset to the Gaussian mean vectors. This approach naturally yields the update formulae for fMPE and motivates an alternative style of training systems. This concept is then extended to the temporal precision matrix modelling (pMPE). In pMPE, a temporally varying positive scale is applied to each element of the diagonal precision matrices. Experimental results are presented on a conversational telephone speech task.

1. Introduction
Hidden Markov Models (HMMs) [1] are the most commonly used acoustic models in state-of-the-art Large Vocabulary Continuous Speech Recognition (LVCSR) [2]. However, HMMs assume that the probability of generating a speech frame given the state is conditionally independent of the previous frames. This is not valid for speech. Trajectory models and switching linear dynamical systems [3] have been proposed to overcome this limitation, but with little success on LVCSR tasks. Recently, fMPE [4] was introduced as a discriminative feature space estimation technique for the HMM-based LVCSR. This method projects a high dimension vector of posteriors down to a standard feature space (typically 39). The parameters of the projection matrix are trained using a gradient-based optimisation of the MPE criterion with an initial Maximum Likelihood (ML) trained model set.

This paper considers IMPE as a form of temporally varying model of the Gaussian mean vectors and extends the concept to the temporal precision matrix modelling (pMPE). In pMPE, a temporally varying positive scale is applied to each element of the diagonal precision matrices. pMPE shares a similar structure of basis interpolation as several existing structured precision matrix approximation schemes [5]. Within the same framework, pMPE can be viewed as modelling the precision matrices by superimposing a set of diagonal basis matrices using some temporally varying weights, which are obtained from the posteriors of the observation vectors given a set of Gaussian components. In addition, this view of temporally varying model parameters motivates an alternative form of system training.

The rest of this paper is organised as follows. Section 2 describes the temporally varying model for the Gaussian mean vectors and the precision matrices. Next, Section 3 derives the estimation formulae for the temporally varying model parameters and discusses several implementation issues. Experimental results are given in Section 4.

2. Temporally Varying Parameters
A time varying mean vector can be expressed as

\[ \mu_{\text{mt}} = \mu_m + b_t = \mu_m + \sum_{i=1}^{n} h_{it} \beta_i, \]  

(1)

where \( b_t \) is a temporally varying shift applied to the original Gaussian mean vectors. This temporally varying shift is given by interpolating the \( n \) basis vectors, \( \beta_i \). The time dependent interpolation weights, \( h_{it} \), are calculated as the posterior probabilities of the feature vector given \( n \) Gaussian components, \( g_i \):

\[ h_{it} = P(g_i | \alpha_t) = \frac{p(\alpha_t | g_i)}{\sum_{j=1}^{n} p(\alpha_t | g_j)} \]  

(2)

where \( p(\alpha_t | g_i) \) is the likelihood of the component \( g_i \) given \( \alpha_t \). This formulation is the same as the fMPE [4] technique, which was viewed as MPE training of the feature space. fMPE estimates a projection matrix from a high dimensional space (∼100,000) down to a standard feature space (typically 39). The columns of the projection matrix corresponds to \( \beta_i \) in equation (1) and the elements of the high dimensional features are given by \( h_{it} \).

A natural extension to the temporally varying mean model is the temporal precision matrix modelling. One possible form, in its most generic expression, is given by

\[ S_{\text{mt}} = A_t' S_m A_t, \]  

(3)

where \( S_m \) and \( S_{\text{mt}} \) are the original and temporal precision matrices. \( A_t \) is a \( d \times d \) time varying transformation matrix:

\[ A_t = I + \sum_{i=1}^{n} h_{it} A_i, \]  

(4)

The expression in equation (3) can be viewed as a temporal Semi-tied Covariance (STC) [6] precision matrix models. However, applying a time varying full transformation matrix, \( A_t \),...
can be computationally expensive. This paper considers a simple form of temporal precision matrix modelling, where diagonal precision matrices and diagonal transforms are used. This simplifies equation (3) to an independent scaling of the diagonal precision matrix elements:

\[ s_{mj} = a_{ij}^2 s_{mj} = \left(1 + \sum_{i=1}^{n} b_{i} a_{ij}\right)^2 s_{mj} \]  

(5)

where \( s_{mj} \) and \( s_{m+j} \) are the static and temporally varying precision of the \( j \)th dimension. \( a_{ij} \) and \( a_{ij} \) are the \( j \)th diagonal element of \( A_i \) and \( A_j \), respectively. The scaling factor at each time is positive to ensure positive-definite precision matrices.

### 3. Parameters Estimation

The model parameters, \( \theta \), can be divided into two sets: static (\( \mu \) and \( S \)) and dynamic (\( b_j \) and \( a_{ij} \)) parameters. This section describes how these parameters can be estimated using the MPE training criterion. In MPE training, the objective function time is positive to ensure positive-definite precision matrices.

**The dynamic model parameters are then estimated using the gradient in equation (9)** where

\[ \frac{\partial Q_{\text{mpe}}}{\partial b_j} = \frac{\partial Q_{\text{mpe}}}{\partial b_j} + \frac{\partial Q_{\text{mpe}}}{\partial \mu_{m+j}} \frac{\partial \mu_{m+j}}{\partial b_j} + \frac{\partial Q_{\text{mpe}}}{\partial \sigma_{m+j}^2} \frac{\partial \sigma_{m+j}^2}{\partial b_j} \]  

(9)

\[ \frac{\partial Q_{\text{mpe}}}{\partial \mu_{m+j}} = h_{m+j} \gamma_{m+j}^2 (\hat{\mu}_{m+j} - \mu_{m+j}) \]  

(10)

\[ \frac{\partial Q_{\text{mpe}}}{\partial \sigma_{m+j}^2} = (y_{m+j} - \hat{\sigma}_{m+j}^2) / 2 \sigma_{m+j}^2 \]  

(11)

The MPE numerator and denominator statistics are calculated in the similar way as the ML statistics given by equations (12) and (13), replacing \( \gamma_{m+j}^2 \) by \( \gamma_{m+j} \) and \( \gamma_{m+j}^2 \) respectively.

### 3.1. Interleaved Dynamic-Static Parameters Estimation

The training method proposed by [4] takes a Maximum Likelihood (ML) trained model and trains the MPE projection matrix. This projection matrix is then used to train the static model parameters using the ML criterion. Repeating this procedure yields an interleaving update for the dynamic and static parameters. The static parameters are updated using the ML criterion by keeping the dynamic parameters constant. The update formulae are derived by maximising the following auxiliary function

\[ Q^m(\theta, \hat{\theta}) = \sum_{t=1}^{T} \sum_{m=1}^{M} \gamma_{m}^2(t) \mathcal{L}^m(\alpha_t) \]  

(10)

where \( \theta \) and \( \hat{\theta} \) denote the set of original and reestimated model parameters respectively, \( \mathcal{L}^m(\alpha_t) \) is the likelihood of the Gaussian component \( m \) given \( \alpha_t \). \( \gamma_{m}^2(t) \) is the posterior of Gaussian component \( m \) at time \( t \). Differentiating equation (10) with respect to \( \mu_{m+j} \) and \( \sigma_{m+j}^2 \) and equating them to zero yield:

\[ \hat{\mu}_{m+j} = x_{m+j}^2 \]  

(12)

\[ \hat{\sigma}_{m+j}^2 = y_{m+j}^2 - x_{m+j}^2 \]  

(13)

\[ \beta_{m+j} = \sum_{t=1}^{T} \gamma_{m}^2(t) \gamma_{m}^2(t) q_{m}^2(t) \]  

(14)

\[ \beta_{m+j} = \sum_{t=1}^{T} \gamma_{m}^2(t) \gamma_{m}^2(t) \gamma_{m}^2(t) q_{m}^2(t) \]  

(15)

\[ \beta_{m+j} = \sum_{t=1}^{T} \gamma_{m}^2(t) \gamma_{m}^2(t) q_{m}^2(t) \gamma_{m}^2(t) q_{m}^2(t) \]  

(16)

The MPE training criterion is realised by maximising the **weak sense** auxiliary function [7]. Standard MPE training of the HMM parameters is realised by maximising the following auxiliary function [7]. \( Q^m(\alpha_t) = \sum_{t=1}^{T} \sum_{m=1}^{M} \gamma_{m}^2(t) \mathcal{L}^m(\alpha_t) \), where

\[ \mathcal{L}^m(\alpha_t) = K_m + \frac{d}{2} \sum_{j=1}^{d} \log s_{mj} - s_{mj} (\alpha_t - \mu_{mj})^2 \]  

(7)

First, consider the update of the dynamic parameters. Due to the large number of posteriori (≈ 100, 000), it is not feasible to accumulate the full second order statistics. Thus, a simple gradient optimisation approach proposed in [4] will be used. For the temporally varying mean model, and MPE, each element of \( \mathbf{b} \) is updated along the gradient of \( Q^m \) with respect to the element, \( b_j \). The gradient is given by

\[ \frac{d Q^m}{db_j} = \frac{d Q^m}{db_j} \]  

(8)

where

\[ Q^m(\alpha_t) = \sum_{t=1}^{T} \gamma_{m}^2(t) \mathcal{L}^m(\alpha_t) \]  

}\[ Q^m(\alpha_t) = \sum_{t=1}^{T} \gamma_{m}^2(t) \mathcal{L}^m(\alpha_t) \]  

(6)

\[ K_m = \frac{1}{2} \sum_{j=1}^{d} \log s_{mj} - s_{mj} (\alpha_t - \mu_{mj})^2 \]  

The last two terms in the right hand side of equation (9) (referred to as the indirect differentials in [4]) also take into account the fact that the global shifting and scaling of the mean should be reflected by updating the static parameters. The actual forms of the differentials \( \frac{\partial Q_{\text{mpe}}}{\partial \mu_{m+j}} \) and \( \frac{\partial Q_{\text{mpe}}}{\partial \sigma_{m+j}^2} \) depend on the update methods for the static parameters, \( \mu_{m+j} \) and \( \sigma_{m+j}^2 \). Ideally, MPE update of the static parameters is preferred. Unfortunately, the use of the \( D \)-smoothing and the \( I \)-smoothing with dynamic ML (or dynamic MMI) priors in standard MPE training [7] complicates the calculation of the indirect differentials. In the following, two simpler forms of update are described.
The dynamic precision matrix parameters, \(a_{ij}\), in pMPE are estimated using a similar gradient-descent based optimisation scheme. Here
\[
\tilde{a}_{ij} = a_{ij} + \eta_i \frac{\partial Q_{mpe}^{pp}}{\partial a_{ij}}
\]
(17)
where \(\tilde{a}_{ij}\) is the updated version of \(a_{ij}\). The element specific learning rate \(\eta_i\) is given by
\[
\eta_i = \alpha \frac{p_{ij} + n_{ij}}{p_{ij}}
\]
(18)
where \(\alpha\) is a scalar parameter for adjusting the learning rate. \(p_{ij}\) and \(n_{ij}\) are the sum of the positive and negative contributions to the gradient at each time, \(t\), computed in a similar way as those for IMPE \([4]\). The gradient is evaluated as
\[
\frac{dQ_{mpe}^{pp}}{da_{ij}} = \sum_{t=1}^{T} \sum_{m=1}^{M} \frac{dQ_{mpe}^{pp}}{da_{ij}}
\]
(19)
where the complete differential of \(Q_{mpe}^{pp}\) with respect to \(a_{ij}\) is given by
\[
\frac{dQ_{mpe}^{pp}}{da_{ij}} = \frac{\partial Q_{mpe}^{pp}}{\partial a_{ij}} + \frac{\partial Q_{mpe}^{pp}}{\partial \mu_{m}^{ij}} \frac{\partial \mu_{m}^{ij}}{\partial a_{ij}} + \frac{\partial Q_{mpe}^{pp}}{\partial \sigma_{m}^{ij}} \frac{\partial \sigma_{m}^{ij}}{\partial a_{ij}}
\]
(20)
and
\[
\frac{\partial Q_{mpe}^{pp}}{\partial a_{ij}} = h_{ij} \gamma_m^{ij}(t)(1 - s_{mij}(a_{ij} - \mu_{mij})^2)
\]
(21)
\[
\frac{\partial Q_{mpe}^{pp}}{\partial \mu_{m}^{ij}} = 2h_{ij} \gamma_m^{ij}(t)(a_{ij} - \mu_{mij})
\]
(22)
\[
\frac{\partial Q_{mpe}^{pp}}{\partial \sigma_{m}^{ij}} = 2h_{ij} \gamma_m^{ij}(t)(a_{ij} - \mu_{mij})^2
\]
(23)
Further MPE training on top of IMPE and pMPE will be called fMPE+MPE and pMPE+MPE respectively.

3.2. Direct Dynamic Parameters Estimation
The estimation method described in 3.1 requires the complete differential to take into account of the change in the model parameters in the subsequent ML training. If only the partial differential is considered, the gain from IMPE and pMPE disappears as soon as the static model parameters are updated \([4]\). However, computing the complete differential requires two passes over the training data. The first pass accumulates the normal MPE statistics \((x_{mij}^{n}, x_{mij}^{s}, y_{mij}^{n}, y_{mij}^{s}, \beta_{m}^{ij} and \beta_{m}^{ij})\) required by equations (15) and (16).

The training time can be reduced if the starting HMM is a well trained MPE model. In this case, the differentials in equations (15) and (16) will have values small enough that can be safely approximated as zero. This conveniently eliminates the need to accumulate the normal MPE statistics. Furthermore, no subsequent reestimation of the static parameters is required. Hence, fMPE and pMPE can be estimated with only a single pass over the training data. These systems are referred to as MPE+IMPE and MPE+MPE respectively.

3.3. Approximate pMPE Training
The mean update in equation (11) requires an additional \(d\)-dimensional vector, \(\beta_{m}^{ij}\) (or \(\beta_{m}^{ij}\)), to be accumulated for each component \(m\). Furthermore, this also complicates the calculation of the D-smoothing constant \([7]\). \(D_m\), for the subsequent MPE training. To simplify the update of the mean vectors, the temporal variation in the scaling factor \(a_{ij}^2\) is ignored when accumulating the mean statistics. Thus, the term \(a_{ij}^2\) in equation (12) may be dropped and \(\beta_{m}^{ij}\) (or \(\beta_{m}^{ij}\) simplifies to \(\beta_{m}^{ij}\) (or \(\beta_{m}^{ij}\)). Since the approximated mean update is independent of \(a_{ij}^2\), \(\beta_{m}^{ij}\) in equation (22) becomes zero. For this approximation to work well, \(a_{ij}\) should be close to the average value of \(a_{ij}\) over time. This approach has been found empirically to yield consistent improvement in both MPE criterion and WER performance, as shown in Section 4.

3.4. Implementation Issues
fMPE has minimal additional cost in terms of likelihood calculation. For pMPE there is a slight increase in this cost. The likelihood of the model parameters, \(\theta^x\) given the observation vector, \(\alpha_t\), is given by equation (7). This requires an extra \(d\) multiplications and 1 addition. It also requires \(a_{ij}\) and \(\sum_{t=1}^{T} \log a_{ij} \) to be cached for each frame, \(t\).

Unlike IMPE, pMPE is more likely to get overtrained, particularly when a higher learning rate is used (\(\alpha > 1.0\)). In such a case, the resulting temporal varying scale, \(a_{ij}^2\) may tend to a value close to zero. To prevent this, a minimum value is applied to \(a_{ij}\), similar to the concept of variance flooring:
\[
\tilde{a}_{ij} = \max(a_{ij}, \alpha_{min})
\]
(24)
where \(\tilde{a}_{ij}\) is the floored scale factor and \(\alpha_{min}\) is the scale floor. In this paper, \(\alpha_{min}\) was set to 0.1.

As mentioned in \([4]\), the update of the dynamic parameters should not result in a global shift or scale in the acoustic space. This provides convenient checks against any implementation errors \([4]\). Similar checks can also be carried out for pMPE implementation by ensuring that the gradient in equation (20) equals zero when there is a global precision scaling. This occurs when \(h_{ij} = h\) is a constant. So,
\[
0 = \sum_{t=1}^{T} \frac{\partial Q_{mpe}^{pp}}{\partial a_{ij}} |_{h_{ij} = h} + \frac{\partial Q_{mpe}^{pp}}{\partial \sigma_{m}^{ij}} \frac{\partial \sigma_{m}^{ij}}{\partial a_{ij}} |_{h_{ij} = h}
\]
(25)
\[
0 = \sum_{t=1}^{T} \frac{\partial Q_{mpe}^{pp}}{\partial a_{ij}} |_{h_{ij} = h} + \frac{\partial Q_{mpe}^{pp}}{\partial \sigma_{m}^{ij}} \frac{\partial \sigma_{m}^{ij}}{\partial a_{ij}} |_{h_{ij} = h}
\]
(26)
Equations (25) and (26) ensure that the dynamic parameters update will not result in a global scaling of the precision matrices.

4. Experimental Results
This section presents the preliminary experimental results of temporal varying Gaussian model on the Conversational Telephone Speech (CTS) English LVCSR task. Systems were trained using the 296 hours switchboard data (bsetselaclin03) and evaluated on a 3-hour test set (clevyn01sub). All systems in this paper used 12 PLP coefficients with the C0 term plus the first, second and third derivatives. An HLDA transformation matrix was used to project the features onto a 39-dimensional space. Side-based cepstral mean, cepstral variance and vocal tract length normalisations were also employed. The baseline system was speaker independent with about 6000 states and 16 Gaussian components per state (\(\sim 99k\) Gaussians in total).

The posteriors, \(h_{ij}\), were calculated based on the same Gaussians in the system. These Gaussians were grouped into 1024 clusters. The posteriors were calculated by evaluating only the Gaussians in the 5 most likely clusters. Posteriors below 0.1 were set to zero to yield approximately 2 non-zero
The larger criterion gain for pMPE did not generalise to recognition increased the criteria of fMPE and pMPE to the ML baseline was 0.74. This was improved by about 0.11 absolute WER reduction respectively. The MPE+fMPE system gave 1.6% and 0.2% absolute better than the MPE system alone. The performance difference between MPE and pMPE+pMPE gradually diminished as the number of MPE training increases. Similar performance was obtained when using the exact pMPE update. However, a slower learning rate (α = 0.5) is required to prevent over training. The gains from MPE and pMPE are not additive. Initial experiment of pMPE training on top of the MPE system (pMPE) showed 0.5% absolute improvement over the MPE system. Unfortunately, this gain decreases with increasing MPE training iterations. More investigation is required to study the interaction between the fMPE and pMPE training.

Table 2 compares the WER performance of MPE+fMPE and MPE+pMPE using the direct estimation scheme. The initial model used by all systems was the MPE system trained with 8 iterations. Four additional standard MPE iterations gave no further improvement. The MPE+fMPE system gave similar performance to the fMPE+MPE system, but the training time for the former is more efficient. Also, four additional direct pMPE training is 0.2% better than the pMPE+MPE system. All the gains over standard MPE presented were statistically significant, except the 0.2% gain from the pMPE+MPE system.

5. Conclusions

This paper has presented a temporal varying model for Gaussian parameters. Applying a temporal varying shift to the mean vector yields the fMPE model. A simple form of temporal precision matrix model is also described. Here the precision matrix is scaled by a temporally varying factor. An alternative training scheme to the standard fMPE is also described. A well trained MPE system is used as the initial model for estimating the dynamic parameters. Both fMPE, temporally varying means, and pMPE, temporally varying precision matrices, yield gains over standard MPE. Future work is required to investigate the interaction between the fMPE and pMPE training.

6. References


1Significance tests were carried out using the NIST Scoring Toolkit