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
{kcs23,mjfg}@eng.cam.ac.uk

Abstract

Many forms of time varying acoustic models have been applied to the area of 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 ($\sim$ 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 fMPE 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 English 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, which is not valid for speech. Trajectory models and switching linear dynamical systems [3] have been proposed to overcome this limitation on small or medium vocabulary systems, but with little success on LVCSR. Recently, fMPE [4] was introduced as a Minimum Phone Error (MPE) training of the feature space 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 model set.

This paper considers fMPE 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

$$\boldsymbol{\mu}_{mt} = \boldsymbol{\mu}_m + \boldsymbol{b}_t = \boldsymbol{\mu}_m + \sum_{i=1}^{n} h_{it}\boldsymbol{b}_i \qquad (1)$$

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

$$h_{it} = P(g_i|\boldsymbol{o}_t) = \frac{p(\boldsymbol{o}_t|g_i)}{\sum_{j=1}^{n} p(\boldsymbol{o}_t|g_j)} \qquad (2)$$

where $p(\boldsymbol{o}_t|g_i)$ is the likelihood of the component g_i given $\boldsymbol{o}_t$. This formulation is the same as the fMPE [4] technique, which was viewed as MPE training of the feature space. This method estimates a projection matrix from a high dimensional space ($\sim$ 100,000) down to a standard feature space (typically 39). The columns of this projection matrix corresponds to $\boldsymbol{b}_i$ 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

$$\boldsymbol{S}_{mt} = \boldsymbol{A}'_t \boldsymbol{S}_m \boldsymbol{A}_t \qquad (3)$$

where $\boldsymbol{S}_m$ and $\boldsymbol{S}_{mt}$ are the original and temporal precision matrices. $\boldsymbol{A}_t$ is a $d \times d$ time varying transformation matrix:

$$\boldsymbol{A}_t = \boldsymbol{I} + \sum_{i=1}^{n} h_{it}\boldsymbol{A}_i \qquad (4)$$

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, $\boldsymbol{A}_t$,

This work was supported by DARPA grant MDA972-02-1-0013. The paper does not necessarily reflect the position or the policy of the US Government and no official endorsement should be inferred.

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:

$$s_{mtj} = a_{tj}^2 s_{mj} = \quad 1 + \sum_{i=1}^{n} h_{it} a_{ij} \Bigg)^2 s_{mj} \qquad (5)$$

where s_{mj} and s_{mtj} are the static and temporally varying precision of the jth dimension. a_{tj} and a_{ij} are the jth diagonal element of $\boldsymbol{A}_t$ and $\boldsymbol{A}_i$ respectively. The scaling factor at each time is positive to ensure positive-definite precision matrices.

3. Parameters Estimation

The model parameters, $\boldsymbol{\theta}$, can be divided into two sets: *static* ($\boldsymbol{\mu}_m$ and $\boldsymbol{S}_m$) and *dynamic* (b_{ij} and a_{tj}) parameters. This section describes how these parameters can be estimated using the MPE training criterion. In MPE training, the objective function to be maximised is given by

$$\mathcal{F}(\boldsymbol{\theta}) = \sum_{r=1}^{R} \sum_{u=1}^{U_r} P_\theta(u|\mathcal{O}_r) A(u, u_r) \qquad (6)$$

where R is the total number of training sentences and U_r is the total number of hypothesised sentences for the rth acoustic data. $A(u, u_r)$ is the *raw phone accuracy* of the sentence u with respect to the reference sentence u_r. $P_\theta(s|\mathcal{O}_r)$ is the posterior sentence probability [7]. Standard MPE training of the HMM parameters is realised by maximising the *weak sense* auxiliary function [7], $\mathcal{Q}^{\texttt{mpe}} = \sum_{t=1}^{T} \sum_{m=1}^{M} \gamma_m^{\texttt{mpe}}(t) \mathcal{L}^m(\boldsymbol{o}_t)$, where

$$\mathcal{L}^m(\boldsymbol{o}_t) = K_m + \frac{1}{2} \sum_{j=1}^{d} \log s_{mtj} - s_{mtj} \left(o_{tj} - \mu_{mtj}\right)^2 \quad (7)$$

K_m subsumes terms independent of the model parameters. T is the total number of training speech frames and M is the total number of Gaussian components in the system. $\gamma_m^{\texttt{mpe}}(t)$ is calculated in normal MPE training [7].

First, consider the update of the dynamic parameters by keeping the static parameters constant. Due to the large number of posteriors ($\sim 100,000$), it is not feasible to accumulate the full second order statistics. Thus, a simple gradient optimisation approach proposed in [4] will be used. For fMPE, each element of $\boldsymbol{b}_i$ is updated along the gradient of $\mathcal{Q}^{\texttt{mpe}}$ with respect to the element, b_{ij}. The gradient is given by

$$\frac{d\mathcal{Q}^{\texttt{mpe}}}{db_{ij}} = \sum_{t=1}^{T} \sum_{m=1}^{M} \frac{d\mathcal{Q}_{mt}^{\texttt{mpe}}}{db_{ij}} \qquad (8)$$

where $\mathcal{Q}_{mt}^{\texttt{mpe}} = \gamma_m^{\texttt{mpe}}(t)\mathcal{L}^m(\boldsymbol{o}_t)$ and

$$\frac{d\mathcal{Q}_{mt}^{\texttt{mpe}}}{db_{ij}} = \frac{\partial\mathcal{Q}_{mt}^{\texttt{mpe}}}{\partial b_{ij}} + \frac{\partial\mathcal{Q}_{mt}^{\texttt{mpe}}}{\partial \mu_{mj}} \frac{\partial \mu_{mj}}{\partial b_{ij}} + \frac{\partial\mathcal{Q}_{mt}^{\texttt{mpe}}}{\partial \sigma_{mj}^2} \frac{\partial \sigma_{mj}^2}{\partial b_{ij}} \qquad (9)$$

Equation (9) is the *complete* differential of $\mathcal{Q}_{mt}^{\texttt{mpe}}$ with respect to b_{ij}. In addition to finding the direction that maximises $\mathcal{Q}_{mt}^{\texttt{mpe}}$, 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 $\frac{\partial\mu_{mj}}{\partial b_{ij}}$ and $\frac{\partial\sigma_{mj}^2}{\partial b_{ij}}$ depend on the update methods for the static parameters, μ_{mj} and σ_{mj}^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.

3.1. Interleaved Dynamic-Static Parameters Estimation

The training method proposed by [4] takes a Maximum Likelihood (ML) trained model and trains the fMPE 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

$$\mathcal{Q}(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}) = \sum_{t=1}^{T} \sum_{m=1}^{M} \gamma_m^{\texttt{ml}}(t) \mathcal{L}^m(\boldsymbol{o}_t) \qquad (10)$$

where $\boldsymbol{\theta}$ and $\hat{\boldsymbol{\theta}}$ denote the set of original and reestimated model parameters respectively. $\mathcal{L}^m(\boldsymbol{o}_t)$ is the likelihood of $\boldsymbol{o}_t$ given the Gaussian component m. $\gamma_m^{\texttt{ml}}(t)$ is the posterior of Gaussian component m at time t. Differentiating equation (10) with respect to μ_{mj} and σ_{mj}^2 and equating them to zero yield:

$$\mu_{mj} = \frac{x_{mj}^{\texttt{ml}}}{\tilde{\beta}_{mj}^{\texttt{ml}}} \quad \text{and} \quad \sigma_{mj}^2 = \frac{y_{mj}^{\texttt{ml}}}{\beta_m^{\texttt{ml}}} \qquad (11)$$

where the required ML statistics are

$$x_{mj}^{\texttt{ml}} = \sum_{t=1}^{T} \gamma_m^{\texttt{ml}}(t) a_{tj}^2 (o_{tj} - b_{tj}) \qquad (12)$$

$$y_{mj}^{\texttt{ml}} = \sum_{t=1}^{T} \gamma_m^{\texttt{ml}}(t) a_{tj}^2 (o_{tj} - \mu_{mtj})^2 \qquad (13)$$

$\beta_m^{\texttt{ml}} = \sum_{t=1}^{T} \gamma_m^{\texttt{ml}}(t)$ and $\tilde{\beta}_{mj}^{\texttt{ml}} = \sum_{t=1}^{T} \gamma_m^{\texttt{ml}}(t) a_{tj}^2$. The sufficient statistics given by equations (12) and (13) are similar to the standard ones except that the observation vectors are shifted by b_{tj} and the occupancy counts, $\gamma_m^{\texttt{ml}}(t)$ are scaled by a_{tj}^2 for each dimension j.

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

$$\frac{\partial\mathcal{Q}_{mt}^{\texttt{mpe}}}{\partial b_{ij}} = \frac{h_{it}\gamma_m^{\texttt{mpe}}(t)(\boldsymbol{o}_t - \mu_{mtj})}{\sigma_{mj}^2} \qquad (14)$$

$$\frac{\partial\mathcal{Q}_{mt}^{\texttt{mpe}}}{\partial \mu_{mj}} = \frac{x_{mj}^{\texttt{n}} - x_{mj}^{\texttt{d}}}{\sigma_{mj}^2} \qquad (15)$$

$$\frac{\partial\mathcal{Q}_{mt}^{\texttt{mpe}}}{\partial \sigma_{mj}^2} = \frac{(y_{mj}^{\texttt{n}} - y_{mj}^{\texttt{d}})/\sigma_{mj}^2 - \beta_m^{\texttt{mpe}}}{2\sigma_{mj}^2} \qquad (16)$$

where the MPE numerator and denominator statistics are calculated in the similar way as the ML statistics given by equations (12) and (13), replacing $\gamma_m^{\texttt{ml}}(t)$ by $\gamma_m^{\texttt{n}}(t)$ and $\gamma_m^{\texttt{d}}(t)$ respectively. $\gamma_m^{\texttt{n}}(t)$ and $\gamma_m^{\texttt{d}}(t)$ are the numerator and denominator occupancy counts given by [7]. $\frac{\partial\mu_{mj}}{\partial b_{ij}}$ and $\frac{\partial\sigma_{mj}^2}{\partial b_{ij}}$ are computed by differentiating the equations in (11) with respect to b_{ij}. For the case where $a_{tj} = 1$, these are the standard fMPE update formulae [4].

The dynamic precision matrix parameters, a_{ij}, in pMPE are estimated using a similar gradient-descent based optimisation scheme. Here

$$\hat{a}_{ij} = a_{ij} + \eta_{ij} \frac{d\mathcal{Q}^{\texttt{mpe}}}{da_{ij}} \tag{17}$$

where $\hat{a}_{ij}$ is the updated version of a_{ij}. The gradient is evaluated as

$$\frac{d\mathcal{Q}^{\texttt{mpe}}}{da_{ij}} = \sum_{t=1}^{T} \sum_{m=1}^{M} \frac{d\mathcal{Q}^{\texttt{mpe}}_{mt}}{da_{ij}} \tag{18}$$

where the complete differential of $\mathcal{Q}^{\texttt{mpe}}_{mt}$ with respect to a_{ij} is given by

$$\frac{d\mathcal{Q}^{\texttt{mpe}}_{mt}}{da_{ij}} = \frac{\partial\mathcal{Q}^{\texttt{mpe}}_{mt}}{\partial a_{ij}} + \frac{\partial\mathcal{Q}^{\texttt{mpe}}_{mt}}{\partial \mu_{mj}} \frac{\partial \mu_{mj}}{\partial a_{ij}} + \frac{\partial\mathcal{Q}^{\texttt{mpe}}_{mt}}{\partial \sigma^2_{mj}} \frac{\partial \sigma^2_{mj}}{\partial a_{ij}} \tag{19}$$

and

$$\begin{aligned}
\frac{\partial\mathcal{Q}^{\texttt{mpe}}_{mt}}{\partial a_{ij}} &= \frac{h_{it}\gamma^{\texttt{mpe}}_m(t)(1 - s_{mtj}(o_{tj} - \mu_{mtj})^2)}{a_{tj}} && (20)\\
\frac{\partial \mu_{mj}}{\partial a_{ij}} &= \frac{2h_{it}\gamma^{\texttt{ml}}_m(t)(o_{tj} - \mu_{mtj})}{\tilde{\beta}^{\texttt{ml}}_{mj}} && (21)\\
\frac{\partial \sigma^2_{mj}}{\partial a_{ij}} &= \frac{2h_{it}a_{tj}\gamma^{\texttt{ml}}_m(t)(o_{tj} - \mu_{mtj})^2}{\beta^{\texttt{ml}}_m} && (22)
\end{aligned}$$

The element specific learning rate η_{ij} is given by

$$\eta_{ij} = \frac{\alpha}{p_{ij} + n_{ij}} \tag{23}$$

where α 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 fMPE [4].

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 fMPE 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 ($x^{\texttt{n}}_{mj}$, $x^{\texttt{d}}_{mj}$, $y^{\texttt{n}}_{mj}$, $y^{\texttt{d}}_{mj}$, $\beta^{\texttt{n}}_m$ and $\beta^{\texttt{d}}_m$) 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.

3.3. Approximating the pMPE Training

The mean update in equation (11) requires an additional d-dimensional vector, $\tilde{\beta}^{\texttt{ml}}_{mj}$ (or $\tilde{\beta}^{\texttt{mpe}}_{mj}$), 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^2_{tj} is ignored when accumulating the mean statistics. Thus, the term a^2_{tj} in equation (12) may be dropped and $\tilde{\beta}^{\texttt{ml}}_{mj}$ (or $\tilde{\beta}^{\texttt{mpe}}_{mj}$) simplifies to $\beta^{\texttt{ml}}_m$ (or $\beta^{\texttt{mpe}}_m$). Since the approximated mean update is independent of a^2_{tj}, $\frac{\partial \mu_{mj}}{\partial s_{mtj}}$ in equation (21) becomes zero. For this approximation to work well, a_{tj} should be *close* to the average value of a_{tj} 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 an observation vector, $\boldsymbol{o}_t$, given the model parameters, $\boldsymbol{\theta}^m$ is given by equation (7) This requires an extra d multiplications and 1 addition require. It also requires a_{tj} and $\sum_{j=1}^{d} \log a_{tj}$ to be cached for each frame, t.

Unlike fMPE, 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^2_{tj} may tend to a value close to zero. To prevent this, a minimum value is asserted onto a_{tj}, similar to the concept of variance flooring:

$$\tilde{a}_{tj} = \max\{a_{tj}, a_{\texttt{min}}\} \tag{24}$$

where $\tilde{a}_{tj}$ is the floored scale factor and $a_{\texttt{min}}$ is the scale floor. In this paper, $a_{\texttt{min}}$ of 0.1 was used.

As mentioned in [4], the update of the dynamic parameters should not result in a *global* shift or scale in the acoustic space, as this should be accounted for by the static parameters. This was achieved by taking the *complete* differentials (including the *indirect* differential). These provide convenient checks against any implementation errors [4]. Similar checks can also be carried out for pMPE implementation by ensuring:

$$\begin{aligned}
0 &= \sum_{t=1}^{T} \frac{\partial\mathcal{Q}^{\texttt{mpe}}_{mt}}{\partial a_{ij}} \,{}_{h_{it}=a_{tj}} + \frac{\partial\mathcal{Q}^{\texttt{mpe}}_{mt}}{\partial \sigma^2_{mj}} \frac{\partial \sigma^2_{mj}}{\partial a_{ij}} \,{}_{h_{it}=a_{tj}} && (25)\\
0 &= \sum_{t=1}^{T} \frac{\partial\mathcal{Q}^{\texttt{mpe}}_{mt}}{\partial \mu_{mj}} \frac{\partial \mu_{mj}}{\partial a_{ij}} \,{}_{h_{it}=a_{tj}} && (26)
\end{aligned}$$

Equations (25) and (26) ensure that there will be no global scaling of the precision matrices. This is also true for the approximation described in Section 3.3 where only equation (25) needs to be checked.

4. Experimental Results

This section presents the preliminary experimental results of temporal varying Gaussian model on LVCSR based on the Conversational Telephone Speech (CTS) English task. Systems are trained using the 296 hours switchboard data (`h5etrain03`) and evaluated on a 3-hour test set (`dev01sub`). All systems in this paper used 12 PLP coefficients with the `C0` term plus the first, second and third derivatives to yield a 52-dimensional feature. A 39×52 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 approximately 6000 states and 16 Gaussian components per state ($\sim$ 99,000 Gaussians in total).

The posteriors, h_{it}, are calculated based on the same Gaussians in the system. These Gaussians are grouped into 1024 clusters. The posteriors are calculated by evaluating only the Gaussians in the 5 most likely clusters. Posteriors below 0.1 are set to zero to yield approximately 2 non-zero posteriors at each

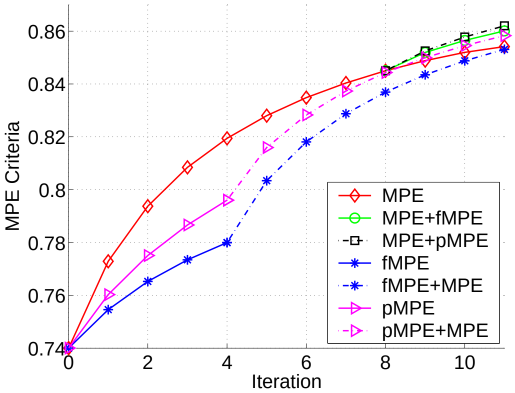

Figure 1: MPE criterion against training iteration

time. No context expansion [4] was used. First, the fMPE and pMPE models were built using four interleaved updates (Section 3.1). 8 MPE iterations were then performed on top of them to give the fMPE+MPE and pMPE+MPE models. MPE+fMPE and MPE+pMPE system were also built using the direct estimation (Section 3.2). The approximation method in Section 3.3 was used for pMPE and pMPE+MPE systems.

Figure 1 shows the change in MPE criteria with increasing training iterations for various systems. The MPE criterion of the ML baseline was 0.74. This was improved by about 0.11 after 12 MPE iterations. The criterion gains for fMPE and pMPE were smaller compared to MPE training. Further MPE training increased the criteria of fMPE+MPE and pMPE+MPE to be similar to the MPE system, with the latter marginally better. The larger criterion gain for pMPE did not generalise to recognition performance (see later). This suggests that pMPE is less robust to overtraining, unlike fMPE [4]. Further criterion gain were obtained with the MPE+fMPE and MPE+pMPE systems.

System	Initial Model	Iter 0	Iter 4	Iter 8
MPE	ML	33.5	30.7	30.2
fMPE+MPE	fMPE	31.9	29.9	29.4
pMPE+MPE	pMPE ($\alpha = 1.0$)	32.5	30.4	30.0

Table 1: WER performance on `dev01sub` for 16-component models using *interleaved* parameters estimation

The Word Error Rate (WER) performance on `dev01sub` for fMPE+MPE and pMPE+MPE systems are shown in Table 1. The ML baseline performance was 33.5%. MPE alone reduced the WER by 3.2% absolute. fMPE and pMPE gave 1.6% and 1.0% absolute WER reduction respectively. Despite the good MPE criterion improvement, the WER performance of the pMPE system converged much quicker (after two iterations). MPE training on top of these systems each gained a further 2.5% absolute, which are respectively 0.8% and 0.2-0.3% absolute better than the MPE system alone. The performance difference between MPE and pMPE+MPE gradually diminished as the number of MPE training increases. The gains from fMPE and pMPE are not *additive*. Initial experiment of pMPE training on top of the fMPE system (fpMPE) showed 0.5% absolute improvement over the fMPE system. Unfortunately, this gain decreases with increasing MPE training iterations. This may be due to the approximation described in Section 3.3. More investigation is required to study the interaction between the fMPE and pMPE training.

System	Iter 0	Iter 2	Iter 4
MPE	30.2	30.2	30.2
MPE+fMPE	30.2	29.6	29.4
MPE+pMPE ($\alpha = 0.5$)	30.2	30.0	29.8

Table 2: WER performance on `dev01sub` for 16-component systems using *direct* parameters estimation

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 presented were found to be statistically significant[1], except the 0.2% gain from the pMPE+MPE system.

5. Conclusions

This paper presented the 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 the between fMPE and pMPE training.

6. References

[1] L. E. Baum and J. A. Eagon, “An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology,” *Bull. Amer. Math. Soc.*, vol. 73, pp. 360–363, 1967.

[2] S. J. Young, “Large vocabulary continuous speech recognition: A review,” in *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*, Snowbird, Utah, December 1995, pp. 3–28.

[3] A-V. I. Rosti and M. J. F. Gales, “Switching linear dynamical systems for speech recognition,” Tech. Rep. CUED/F-INFENG/TR461, Cambridge University, 2003.

[4] D. Povey, B. Kingsbury, L. Mangu, G. Saon, H. Soltau, and G. Zweig, “fMPE: Discriminatively trained features for speech recognition,” in *Proc.ICASSP*, 2005.

[5] K C Sim and M J F Gales, “Precision matrix modelling for large vocabulary continuous speech recognition,” Tech. Rep. CUED/F-INFENG/TR485, Cambridge University, 2004.

[6] M. J. F. Gales, “Semi-tied covariance matrices for hidden Markov models,” *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 3, pp. 272–281, 1999.

[7] D. Povey and P. C. Woodland, “Minimum Phone Error and I-smoothing for improved discriminative training,” in *Proc.ICASSP*, 2002.

[1] Significance tests were carried out using the NIST Scoring Toolkit