PARALLEL MODEL COMBINATION
FOR SPEECH RECOGNITION IN NOISE

M. J. F. Gales & S. J. Young

CUED/F-INFENG/TR 135

June 1993

Cambridge University Engineering Department
Trumpington Street
Cambridge CB2 1PZ
England

Email: mjfg@eng.cam.ac.uk

Abstract

This report addresses the problem of automatic speech recognition in the presence of interfering noise. The approach adopted is to compensate the parameters of a clean speech model given the statistics of the interfering noise. In this work these statistics are assumed to be modelled by a Hidden Markov Model (HMM). The basic theory of static coefficient Parallel Model Combination (PMC) is reviewed and placed within the framework of approximating the Maximum Likelihood (ML) estimate of the corrupted speech model, given the clean speech and interfering noise models. In addition the paper examines the problem of compensating delta coefficients in a PMC framework. Expressions for ML estimates of delta coefficients are derived and computationally efficient approximations of these estimates are given. The effectiveness of compensating delta parameters is discussed.

Keywords: speech recognition, noise compensation, HMM, PMC.

Contents

1 Introduction

As speech recognition technology moves from the laboratory to real applications, there is a need to make systems which are robust to a wide variety of background noises. Many different approaches to achieving noise robustness have been studied [12]. These approaches may be split into two groups.

Firstly, the corrupted waveform may be preprocessed in such a way that the resulting parameters are closely related to those of clean speech. Techniques in this category include spectral subtraction [13, 11] spectral mapping [7] and inherently robust parameterisations [5]. These methods only use statistical information about the interfering noise in the compensation process, no account is made of what was said.

The second class of methods attempt to modify the pattern matching stage in order to account for the interfering noise. Methods in this approach include noise masking [6, 4], state based filtering [15], cepstral mean compensation [1, 16] and HMM decomposition [3].

This paper is concerned with the latter approach to noise robustness. In particular the scheme based on Parallel Model Combination (PMC) [9, 10]. The basic concept behind PMC assumes that the performance of speech recognition systems is optimum when there is no mismatch between training and test conditions. Specifically, PMC considers the case where there is interfering additive noise. Invariably in real applications, there is some mismatch between training and test conditions, so some method for compensating the parameters of the models, or re-training of the models is required. If the effect of the 'mismatch' is known, for example in interfering additive noise, it should be possible to modify the training data to match this new test condition and then re-train the models. This would require that the whole database be stored and modified whenever the conditions change, a highly computationally expensive task. It is therefore necessary to compress the training data into a more manageable form. One method is to store statistics derived from the training data. The task is then to train a new set of models using these training statistics. To this end the standard HMM re-estimation formulae are modified to accommodate statistical training data. The modifications are described in section 2, where re-estimation formulae for Maximum Likelihood (ML) estimates of the parameters are given. The theory is then applied to the specific case where the training data is modelled using HMMs.

In section 3 the theory of PMC is reviewed and placed within the concept of making a ML estimate of corrupted speech models where there is interfering additive noise. Here, both the speech and the interfering noise are required to be modelled. As previously stated, the speech may be modelled using standard HMMs. If there is significant temporal information in the interfering noise, it is also necessary to model the noise using an HMM. Computationally highly efficient approximations to the true ML estimates are derived.

The basic theory of PMC is extended to include delta, differential, parameters in section 4. If model compensation techniques are to be applied to large vocabulary tasks, where it is essential to use dynamic coefficients to achieve good recognition performance, methods for compensating the dynamic coefficients must be found. Firstly, it is necessary to find what effect the 'mismatch' has on delta parameters. Approximations are then made of this mismatch function which allow the model parameters to be compensated using the statistics obtained from standard HMMs. Again computationally efficient approximations are derived.

Section 5 details the experimental work carried out on the NOISEX-92 database. Firstly the performance of the static coefficient compensation schemes is examined. Both the approximate scheme and numerical integration of the exact values are compared to uncompensated models and models trained under the noise conditions. The performance of delta coefficient compensation is then examined.

2 Training HMMs on Statistical Data

It is well known that the performance of HMM based speech recognition systems degrades rapidly as the mismatch between the training and test conditions increases. A scheme which modifies the parameters of the models to compensate for this mismatch in conditions is therefore required. If the compensation process is to be rapid it is normally not possible to compensate the whole of the training database for the 'mismatch' and then re-train the models. This would firstly require that all the training data is available on line and that sufficient time is available to compensate it all. Both are unlikely under real test conditions. An alternative is to use statistics derived from the training data. If all the information from the training data is accurately represented in these statistics and the 'mismatch' correctly modelled, there will be no degradation in performance compared to training and testing under the same conditions. Assuming that such statistics exist, it is necessary to modify the standard HMM re-estimation formulae to accommodate the use of statistical data, instead of actual observations. Firstly an optimisation criterion must be chosen. For this work a Maximum Likelihood estimate will be used. The notation adopted in this section of the report is based on that used in the HTK Manual [14]. A more detailed discussion of HMM theory and application for speech recognition is given by Rabiner [8]. Looking at the standard re-estimation formula of the new mean of mixture M_m of state S_j, $\hat{\mu}_{jm}$

$$\hat{\mu}_{jm} = \frac{\sum_{\tau=1}^{T} L_{jm}(\tau)\mathbf{y}_\tau}{\sum_{\tau=1}^{T} L_{jm}(\tau)} \tag{1}$$

where

$$L_{jm}(\tau) = \frac{1}{P} U_j(\tau) c_{jm} \beta_j(\tau) p(\mathbf{y}_\tau | S_j, M_m, \mathcal{M}) \tag{2}$$

and

$$U_j(\tau) = \begin{cases} a_{1j} & (\text{if } \tau = 1) \\ \sum_{i=2}^{N-1} \alpha_i(\tau-1) a_{ij} & (\text{otherwise}) \end{cases} \tag{3}$$

c_{jm} is the mixture weight associated with mixture M_m of state S_j,

$$a_{ij} = p(q_j(t+1)|q_i(t)) \tag{4}$$

where $q_i(t)$ indicates state S_i at time t and

$$\alpha_i(t) = p(\mathbf{y}_1, .., \mathbf{y}_t, q_i(t)|\mathcal{M}) \tag{5}$$

$$\beta_i(t) = p(\mathbf{y}_{t+1}, .., \mathbf{y}_T|q_i(t), \mathcal{M}) \tag{6}$$

This can then be expressed in terms of the expected values of random variables instead of true observations. As $T \to \infty$ equation 1 can be rewritten

$$\hat{\mu}_{jm} = \frac{T\mathcal{E}\{L_{jm}(\tau)\mathbf{y}_\tau\}}{T\mathcal{E}\{L_{jm}(\tau)\}} = \frac{\mathcal{E}\{L_{jm}(\tau)\mathbf{y}_\tau\}}{\mathcal{E}\{L_{jm}(\tau)\}} \tag{7}$$

Similar expressions for the variance and mixture weights in terms of expected values may be obtained.

So far nothing has been assumed about the form of the statistics. Given the nature of the speech signal, highly correlated data with temporal information, it is not possible to accurately store all the information about the database in a small number of statistical parameters. If this were possible, the task of speech recognition would be far easier. However, it is possible to store considerable information about the speech using HMMs. HMMs have been shown to achieve good recognition performance and hence may be assumed to contain sufficient information about the training data for recognition purposes. Furthermore, there exist elegant and efficient training algorithms for HMMs. If HMMs are used to model the training data and the frame state allocation is not altered then

$$\frac{1}{P} U_j(\tau) \beta_j(\tau) p(\mathbf{y}_\tau | S_j, \mathcal{M}) = 1 \tag{8}$$

So

$$L_{jm}(\tau) = p(M_m|\mathbf{y}_\tau, S_j, \mathcal{M}) = \frac{c_m \mathcal{N}(\mathbf{y}_\tau; \mu_{jm}, \mathbf{\Sigma}_{jm})}{\sum\limits_{i=1}^{M} c_i \mathcal{N}(\mathbf{y}_\tau; \mu_{ji}, \mathbf{\Sigma}_{ji})} = \mathcal{K}_m(\mathbf{y}_\tau) \tag{9}$$

For multiple Gaussian mixture HMMs the re-estimation formula becomes

$$\hat{\mu}_{jm} = \frac{\int\limits_{\mathcal{R}^n} \mathcal{K}_m(\mathbf{y})\mathbf{y}p(\mathbf{y})\,d\mathbf{y}}{\int\limits_{\mathcal{R}^n} \mathcal{K}_m(\mathbf{y})p(\mathbf{y})\,d\mathbf{y}} \tag{10}$$

Similarly the re-estimation for the covariance matrix becomes

$$\begin{aligned}\hat{\mathbf{\Sigma}}_{jm} &= \frac{\int\limits_{\mathcal{R}^n} \mathcal{K}_m(\mathbf{y})(\mathbf{y}-\hat{\mu}_{jm})(\mathbf{y}-\hat{\mu}_{jm})^T p(\mathbf{y})\,d\mathbf{y}}{\int\limits_{\mathcal{R}^n} \mathcal{K}_m(\mathbf{y})p(\mathbf{y})\,d\mathbf{y}} \\ &= \left(\frac{\int\limits_{\mathcal{R}^n} \mathcal{K}_m(\mathbf{y})\mathbf{y}\mathbf{y}^T p(\mathbf{y})\,d\mathbf{y}}{\int\limits_{\mathcal{R}^n} \mathcal{K}_m(\mathbf{y})p(\mathbf{y})\,d\mathbf{y}}\right) - \hat{\mu}_{jm}\hat{\mu}_{jm}^T \end{aligned} \tag{11}$$

and for the weights

$$\hat{c}_{jm} = \int\limits_{\mathcal{R}^n} \mathcal{K}_m(\mathbf{y})p(\mathbf{y})\,d\mathbf{y} \tag{12}$$

The transition matrix will remain the same as the frame state allocation is assumed to be unaltered. All the above expressions may be seen to be functions of

$$\mathcal{E}\left\{\mathcal{K}_m(\mathbf{y})\right\} = \int\limits_{\mathcal{R}^n} \mathcal{K}_m(\mathbf{y})p(\mathbf{y})\,d\mathbf{y} \tag{13}$$

$$\mathcal{E}\left\{\mathbf{y}\mathcal{K}_m(\mathbf{y})\right\} = \int\limits_{\mathcal{R}^n} \mathbf{y}\mathcal{K}_m(\mathbf{y})p(\mathbf{y})\,d\mathbf{y} \tag{14}$$

$$\mathcal{E}\left\{\mathbf{y}\mathbf{y}^T\mathcal{K}_m(\mathbf{y})\right\} = \int\limits_{\mathcal{R}^n} \mathbf{y}\mathbf{y}^T\mathcal{K}_m(\mathbf{y})p(\mathbf{y})\,d\mathbf{y} \tag{15}$$

If a single Gaussian mixture model is to be estimated, $M = 1$, then

$$\mathcal{K}_m(\mathbf{y}) = \mathcal{K}(\mathbf{y}) = 1 \tag{16}$$

and the formulae may be simplified accordingly.

No assumptions have been made in the above analysis about the form of the probability distribution of $\mathbf{y}$. If it is derived from a standard HMM then it may either be a single or multiple Gaussian mixture. For both single and multiple Gaussian mixtures the analysis is valid. It is not necessary to assume that the frame state allocation is fixed, however given the poor assumptions of the HMM with regards to most real signals and the poor durational modelling of the standard HMM, little will be gained by relaxing this assumption. If multiple Gaussian mixtures are to be estimated it is necessary to iterate in a standard *EM* fashion to obtain the ML estimate, unless the same number of states is to used in the compensated models as were in the training statistics and the form of $\mathbf{y} = \mathcal{F}()$, where $\mathcal{F}()$ describes the effect of the mismatch on the clean parameters, is linear, or a single mixture is to be used.

3 Basic PMC Theory

The objective of any HMM based noise compensation scheme is to estimate, according to some objective function, the corrupted speech model given information about the clean speech and interfering noise. If the corrupted speech is to be modelled by a standard HMM, then to obtain the ML estimate of the noise compensated speech model it is necessary to estimate the mean and covariance of the corrupted speech. This is a specific example of training HMMs on statistical data, as described in the previous section. For PMC the mismatch is caused by interfering additive noise. It is necessary to define the 'mismatch' function for this case. To obtain this function a series of assumptions are made.

1. The speech and noise are independent.
2. The speech and noise are additive in the linear domain. In addition it is assumed that there is sufficient smoothing on the spectral estimate so that the speech and noise may be assumed to be additive at the power spectrum level.
3. A single Gaussian or set of Gaussian mixtures contain sufficient information to represent the distribution of the observation vectors in the log domain.
4. The frame state allocation is not altered by the addition of noise.

The 'observations' are then given by the 'mismatch' function

$$\mathbf{y}_i(t) = \mathbf{O}^l_i(t) = \mathcal{F}(\mathbf{S}^l_i(t), \mathbf{N}^l_i(t)) = \log(g\exp(\mathbf{S}^l_i(t)) + \exp(\mathbf{N}^l_i(t))) \tag{17}$$

where g is a gain matching term introduced to account for level differences between the clean speech and the noisy speech, $\mathbf{S}^l(t)$ is the clean speech and $\mathbf{N}^l(t)$ is the interfering noise. Throughout the rest of this paper the superscript will be used to indicate the domain of the variable. Thus $\mathbf{O}^c(t)$ is the corrupted speech observation in the cepstral domain, $\mathbf{O}^l(t)$ is in the log spectrum domain and $\mathbf{O}(t)$ is in the linear spectrum domain. Furthermore $\mathbf{O}^c(t)$ will represent the observation at time t, the associated random variable will be $\mathbf{O}^c$. All variables in bold are vectors or matrices, subscripts indicating elements of the vector or matrix. In the above expression a log() compression function has been used, as is normal with the use of cepstral coefficients. An appropriately modified 'mismatch' function may be used with any compression function, however, the approximations used in this section are specific to the log() compression. For simplicity of notation only one state of each model will be considered. The way in which multi-state models are combined is detailed in previous work [10].

Substituting equation 17 in equation 10, the new estimate of the mean is

$$\hat{\mu}^l_i(m) = \frac{\int_{\mathcal{R}^n} d\mathbf{S}^l \int_{\mathcal{R}^n} d\mathbf{N}^l \; [\mathcal{K}_m(\mathbf{S}^l, \mathbf{N}^l)\log(g\exp(\mathbf{S}^l_i) + \exp(\mathbf{N}^l_i))p(\mathbf{S}^l)p(\mathbf{N}^l)]}{\int_{\mathcal{R}^n} d\mathbf{S}^l \int_{\mathcal{R}^n} d\mathbf{N}^l \; [\mathcal{K}_m(\mathbf{S}^l, \mathbf{N}^l)p(\mathbf{S}^l)p(\mathbf{N}^l)]} \tag{18}$$

where $\hat{\mu}^l_i(m)$ is the i^{th} element of the mean associated with the m^{th} mixture. $\mathcal{F}(\mathbf{S}^l_i, \mathbf{N}^l_i)$ is non-linear so if multiple mixture models are to be estimated then it is necessary to use an iterative scheme. If only a single mixture HMM is to be estimated then $\mathcal{K}_m(\mathbf{S}^l, \mathbf{N}^l) = 1$ and the above expression may be simplified to

$$\hat{\mu}^l_i = \int_{\mathcal{R}^n} d\mathbf{S}^l \int_{\mathcal{R}^n} d\mathbf{N}^l \; \log(g\exp(\mathbf{S}^l_i) + \exp(\mathbf{N}^l_i))p(\mathbf{S}^l)p(\mathbf{N}^l) \tag{19}$$

Throughout the rest of this report only single Gaussian mixture models will be considered, so it is not necessary to use an iterative scheme to obtain the ML estimate. The covariance estimation formula can be written in the same form. Substituting equation 17 in equation 11 yields

$$\hat{\mathbf{\Sigma}}^l_{ij} = \int_{\mathcal{R}^n} d\mathbf{S}^l \int_{\mathcal{R}^n} d\mathbf{N}^l \log(g\exp(\mathbf{S}^l_i) + \exp(\mathbf{N}^l_i))\log(g\exp(\mathbf{S}^l_j) + \exp(\mathbf{N}^l_j))p(\mathbf{S}^l)p(\mathbf{N}^l) - \hat{\mu}^l_i\hat{\mu}^l_j \tag{20}$$

There are no weights to re-estimate.

If the speech and noise are modelled by separate HMMs trained on cepstral feature vectors having Gaussian distributions with parameters $\{\mu^c, \mathbf{\Sigma}^c\}$ and $\{\tilde{\mu}^c, \tilde{\mathbf{\Sigma}}^c\}$ respectively, it is necessary to map these parameters to the log spectrum domain. For the speech

$$\mu^l = \mathbf{C}^{-1}\mu^c \tag{21}$$
$$\mathbf{\Sigma}^l = \mathbf{C}^{-1}\mathbf{\Sigma}^c(\mathbf{C}^{-1})^T \tag{22}$$

and similarly the noise parameters $\{\tilde{\mu}^c, \tilde{\mathbf{\Sigma}}^c\}$ may be mapped to $\{\tilde{\mu}^l, \tilde{\mathbf{\Sigma}}^l\}$ where $\mathbf{C}$ is the matrix representing the discrete cosine transform. No additional assumptions are required at this stage, as the linear combination of Gaussian distributed random variables is itself Gaussian distributed.

There is no closed form for either the compensated mean, $\hat{\mu}^l$ or covariance, $\hat{\mathbf{\Sigma}}^l$, as described by equations 19 and 20. So to obtain exact forms for the above expressions would require multi-dimensional numerical integration. However, if it is assumed that the sum of two lognormally distributed variables is itself approximately lognormally distributed then it is only necessary to calculate the mean and the variance in the linear spectrum domain. Given the previously stated assumptions that the speech and noise are independent and additive in the linear spectrum domain

$$\hat{\mu} = g\mu + \tilde{\mu} \tag{23}$$
$$\hat{\mathbf{\Sigma}} = g^2\mathbf{\Sigma} + \tilde{\mathbf{\Sigma}} \tag{24}$$

where the parameter set $\{\mu, \mathbf{\Sigma}\}$ are the mean and covariance respectively of the lognormal distribution associated with the Gaussian distribution $\{\mu^l, \mathbf{\Sigma}^l\}$ and similarly $\{\tilde{\mu}, \tilde{\mathbf{\Sigma}}\}$ and $\{\tilde{\mu}^l, \tilde{\mathbf{\Sigma}}^l\}$. The parameters of the clean speech in the linear and log spectrum domains are related by

$$\mu_i = \exp(\mu_i^l + \mathbf{\Sigma}_{ii}^l/2) \tag{25}$$
$$\mathbf{\Sigma}_{ij} = \mu_i\mu_j\left[\exp(\mathbf{\Sigma}_{ij}^l) - 1\right] \tag{26}$$

and similarly for the noise [10]. As the corrupted speech is assumed to be lognormally distributed in the linear spectrum domain, the required distribution in the log spectrum domain, $\{\hat{\mu}^l, \hat{\mathbf{\Sigma}}^l\}$, may be obtained using the inverse of the above expressions. Hence

$$\hat{\mu}_i^l = \log(\hat{\mu}_i) - \frac{1}{2}\log\left(\frac{\hat{\mathbf{\Sigma}}_{ii}}{\hat{\mu}_i^2} + 1\right) \tag{27}$$
$$\hat{\mathbf{\Sigma}}_{ij}^l = \log\left(\frac{\hat{\mathbf{\Sigma}}_{ij}}{\hat{\mu}_i\hat{\mu}_j} + 1\right) \tag{28}$$

If cepstral parameters are to be used in the recognition stage, then we use the final mapping

$$\mu^c = \mathbf{C}\mu^l \tag{29}$$
$$\mathbf{\Sigma}^c = \mathbf{C}\mathbf{\Sigma}^l\mathbf{C}^T \tag{30}$$

This process can be viewed as an approximation to the ML estimate of a Gaussian distribution of the observed corrupted speech signal, $\mathbf{O}^c(t)$ given the Gaussian distributions of the clean speech and interfering noise, with minimal computational overhead.

4 Delta Coefficient Compensation Theory

For large vocabulary speech recognition it is necessary to incorporate dynamic coefficients in the speech parameterisation to achieve good recognition performance. The basic theory for PMC has relied on the fact that the speech and noise are additive. Hence the corrupted speech signal is a simple combination of the speech and noise signal in the linear spectrum domain. When dynamic coefficients are used this simple combination is not possible. To implement delta coefficient compensation within the PMC framework it is necessary to obtain the new 'mismatch' function. If the speech is now parameterised using

$$\mathbf{O}^{\Delta c}(t)^T = \left[\mathbf{O}^c(t)^T, \mathbf{\Delta O}^c(t)^T\right] \tag{31}$$

where $\mathbf{\Delta O}^c(t)$ are the simplest form of dynamic coefficients, delta coefficients, then

$$\begin{aligned}\mathbf{\Delta O}^c(t) &= (\mathbf{O}^c(t+1) - \mathbf{O}^c(t-1)) \\ &= \mathbf{C}(\mathbf{O}^l(t+1) - \mathbf{O}^l(t-1)) \\ &= \mathbf{C}\log\left[(\mathbf{O}(t+1))./(\mathbf{O}(t-1))\right]\end{aligned} \tag{32}$$

where ./ is elementwise division. Using the assumption that the speech and noise are additive, $\mathbf{O}(t) = \mathbf{S}(t) + \mathbf{N}(t)$, and substituting this in the above equation

$$\mathbf{\Delta O}^c(t) = \mathbf{C}\log\left[(\mathbf{S}(t+1) + \mathbf{N}(t+1))./(\mathbf{S}(t-1) + \mathbf{N}(t-1))\right] \tag{33}$$

This may be expressed in terms of the delta coefficients of the speech, $\mathbf{\Delta S}(t)$, and noise, $\mathbf{\Delta N}(t)$, in the linear domain

$$\begin{aligned}\mathbf{\Delta O}_i(t) &= \left(\frac{\mathbf{S}_i(t+1)}{\mathbf{S}_i(t-1) + \mathbf{N}_i(t-1)}\right) + \left(\frac{\mathbf{N}_i(t+1)}{\mathbf{S}_i(t-1) + \mathbf{N}_i(t-1)}\right) \\ &= \mathbf{\Delta S}_i(t)\left(\frac{\frac{\mathbf{S}_i(t-1)}{\mathbf{N}_i(t-1)}}{\frac{\mathbf{S}_i(t-1)}{\mathbf{N}_i(t-1)} + 1}\right) + \mathbf{\Delta N}_i(t)\left(\frac{1}{\frac{\mathbf{S}_i(t-1)}{\mathbf{N}_i(t-1)} + 1}\right)\end{aligned} \tag{34}$$

The corrupted speech cepstral delta coefficients have been rewritten in terms of the static and delta coefficients of the clean speech and interfering noise. Examining the observation time the delta coefficient at time t is dependent on the static coefficients at time $t-1$. This is contrary to one of the assumptions behind the use of HMMs for speech recognition, that the speech waveform may be split into stationary segments with instantaneous transitions between them. However, if the segments are assumed to be long enough then the statistics of $\mathbf{S}(t-1)$ will be approximately the same as those of $\mathbf{S}(t)$ and $\mathbf{N}(t-1)$ the same as $\mathbf{N}(t)$. With this assumption statistics exist for all the variables of equation 34. This is then an appropriate 'mismatch' function.

An ML estimate of the delta parameters of the HMM is needed. Again this requires the mean and the covariance of the signal in the log spectrum domain to be calculated. If the speech is parameterised in the cepstral domain it must be mapped to the log spectrum domain. For the speech

$$(\mu^{\Delta l})^T = \left[(\mathbf{C}^{-1}\mu^c)^T, (\mathbf{C}^{-1}\mathbf{\Delta}\mu^c)^T\right] \tag{35}$$

and

$$\mathbf{\Sigma}^{\Delta l} = \left[\begin{array}{cc}\mathbf{C}^{-1}\mathbf{\Sigma}^c(\mathbf{C}^{-1})^T & \mathbf{C}^{-1}\delta\mathbf{\Sigma}^c(\mathbf{C}^{-1})^T \\ \mathbf{C}^{-1}(\delta\mathbf{\Sigma}^c)^T(\mathbf{C}^{-1})^T & \mathbf{C}^{-1}\mathbf{\Delta\Sigma}^c(\mathbf{C}^{-1})^T\end{array}\right] \tag{36}$$

where $\delta\mathbf{\Sigma}^c$ is the covariance matrix representing the correlation between the static and delta coefficients. A similar mapping converts the noise parameters $\{\tilde{\mu}^{\Delta c}, \tilde{\mathbf{\Sigma}}^{\Delta c}\}$ to $\{\tilde{\mu}^{\Delta l}, \tilde{\mathbf{\Sigma}}^{\Delta l}\}$. The ML estimates of the static coefficients are unaltered. Those for the delta coefficients are given by substituting equation 34 in equations 10 and 11 and assuming a single gaussian mixture is to be estimated.

$$\mathbf{\Delta}\hat{\mu}_i^l = \int_{\mathcal{R}^n} d\mathbf{S}^l \int_{\mathcal{R}^n} d\mathbf{N}^l \int_{\mathcal{R}^n} d\mathbf{\Delta S}^l \int_{\mathcal{R}^n} d\mathbf{\Delta N}^l p(\mathbf{S}^l, \mathbf{\Delta S}^l) p(\mathbf{N}^l, \mathbf{\Delta N}^l) \log\left(\gamma_i \exp(\mathbf{\Delta S}_i^l) + \eta_i \exp(\mathbf{\Delta N}_i^l)\right) \tag{37}$$

and

$$\begin{aligned}\mathbf{\Delta}\hat{\mathbf{\Sigma}}_{ij}^l = & \int_{\mathcal{R}^n} d\mathbf{S}^l \int_{\mathcal{R}^n} d\mathbf{N}^l \int_{\mathcal{R}^n} d\mathbf{\Delta S}^l \int_{\mathcal{R}^n} d\mathbf{\Delta N}^l \\ & \left\{p(\mathbf{S}^l, \mathbf{\Delta S}^l) p(\mathbf{N}^l, \mathbf{\Delta N}^l) \log\left(\gamma_i \exp(\mathbf{\Delta S}_i^l) + \eta_i \exp(\mathbf{\Delta N}_i^l)\right) \log\left(\gamma_j \exp(\mathbf{\Delta S}_j^l) + \eta_j \exp(\mathbf{\Delta N}_j^l)\right)\right\} \\ & - \mathbf{\Delta}\hat{\mu}_i^l \mathbf{\Delta}\hat{\mu}_j^l\end{aligned} \tag{38}$$

and

$$\delta \hat{\boldsymbol{\Sigma}}^l_{ij} = \int_{\mathcal{R}^n} d\mathbf{S}^l \int_{\mathcal{R}^n} d\mathbf{N}^l \int_{\mathcal{R}^n} d\boldsymbol{\Delta}\mathbf{S}^l \int_{\mathcal{R}^n} d\boldsymbol{\Delta}\mathbf{N}^l \tag{39}$$

$$\left\{ p(\mathbf{S}^l, \boldsymbol{\Delta}\mathbf{S}^l) p(\mathbf{N}^l, \boldsymbol{\Delta}\mathbf{N}^l) \log\left(g \exp(\mathbf{S}^l_i) + \exp(\mathbf{N}^l_i)\right) \log\left(\gamma_j \exp(\boldsymbol{\Delta}\mathbf{S}^l_j) + \eta_j \exp(\boldsymbol{\Delta}\mathbf{N}^l_j)\right) \right\}$$
$$-\hat{\mu}^l_i \boldsymbol{\Delta}\hat{\mu}^l_j$$

where

$$\gamma_i = \left(\frac{\exp(\mathbf{S}^l_i - \mathbf{N}^l_i)}{\exp(\mathbf{S}^l_i - \mathbf{N}^l_i) + 1} \right) \tag{40}$$

$$\eta_i = \left(\frac{1}{\exp(\mathbf{S}^l_i - \mathbf{N}^l_i) + 1} \right) \tag{41}$$

If diagonal covariance matrices for the corrupted speech are to be estimated it is not necessary to estimate $\delta \hat{\boldsymbol{\Sigma}}^l$. To calculate the full forms of equation 37 and equation 38 again requires multi-dimensional numerical integration, which is computationally expensive. However by making an additional assumption that the variances on γ and η are negligible then the form of the ML estimates of the delta parameters are the same as those of the static coefficients. Hence

$$\boldsymbol{\Delta}\hat{\mu}_i = \overline{\gamma}_i \boldsymbol{\Delta}\mu_i + \overline{\eta}_i \boldsymbol{\Delta}\tilde{\mu}_i \tag{42}$$

$$\boldsymbol{\Delta}\hat{\boldsymbol{\Sigma}}_{ij} = \overline{\gamma}_i \overline{\gamma}_j \boldsymbol{\Delta}\boldsymbol{\Sigma}_{ij} + \overline{\eta}_i \overline{\eta}_j \boldsymbol{\Delta}\tilde{\boldsymbol{\Sigma}}_{ij} \tag{43}$$

where

$$\mathcal{E}\left[\gamma_i\right] = \mathcal{E}\left[\frac{\frac{\mathbf{S}_i}{\mathbf{N}_i}}{\frac{\mathbf{S}_i}{\mathbf{N}_i} + 1} \right] \approx \left(\frac{\frac{\mu_i}{\tilde{\mu}_i}}{\frac{\mu_i}{\tilde{\mu}_i} + 1} \right) = \overline{\gamma}_i \tag{44}$$

and

$$\mathcal{E}\left[\eta_i\right] = \mathcal{E}\left[\frac{1}{\frac{\mathbf{S}_i}{\mathbf{N}_i} + 1} \right] \approx \left(\frac{1}{\frac{\mu_i}{\tilde{\mu}_i} + 1} \right) = \overline{\eta}_i \tag{45}$$

The mean and covariance can now be mapped back into the cepstral domain in a similar way to the static coefficients

$$\boldsymbol{\Delta}\mu^c = \mathbf{C}\boldsymbol{\Delta}\mu^l \tag{46}$$

$$\boldsymbol{\Delta}\boldsymbol{\Sigma}^c = \mathbf{C}\boldsymbol{\Delta}\boldsymbol{\Sigma}^l\mathbf{C}^T \tag{47}$$

5 Evaluation on NOISEX-92

In this section a number of experiments using the NOISEX-92 database [2] are reported. The data was preprocessed using a 25 msec Hamming window and a 10 msec frame period. For each frame a set of 15 MFCC were computed. The zeroth cepstral coefficients is computed and stored since it is needed in the PMC mapping process. Where delta coefficients are used all 15 delta MFCC are calculated.

For each digit, a single mixture continuous density HMM with 8 emitting states was trained using the clean data only. The topology for all models was left-right with no skips and diagonal covariance matrices were assumed throughout. For each test condition, a single state diagonal covariance noise HMM was trained using the silence intervals of the test files. Recognition used a standard connected word Viterbi Decoder constrained by a syntax consisting of silence followed by a digit in a loop. Thus no explicit end-point detector was used and insertion/deletion errors occurred as well as classification errors. The results are in terms of % accuracy where for N tokens, S substitution errors, D deletion errors and I insertion errors, accuracy is calculated as $[(N - S - D - I)/N] \times 100\%$. The error counts themselves were calculated by using a DP string matching algorithm between the recognised digit sequence and the reference transcription. All training and testing used version 1.4 of the portable HTK HMM toolkit [14] with suitable extensions to perform PMC.

SNR	*Lynx*		*F16*		*Car*	
dB	Clean	Noisy	Clean	Noisy	Clean	Noisy
-06	12	54	12	50	23	79
+00	25	98	17	86	32	96
+06	59	100	50	98	75	98
+12	97	100	82	100	96	100
+18	100	100	95	100	99	100

Table 1: Baseline Static Performance

The initial set of experiments were performed to examine the performance of PMC compared with the training and testing under the same condition. Table 1 shows the results for uncompensated model parameters, *Clean*, and models trained under the test conditions, *Noisy*. For the *Noisy* models the same complete dataset was used as for the clean training. This should therefore be the best possible performance obtainable using model compensation techniques.

SNR	*Lynx*		*F16*		*Car*	
dB	Fast	Num.	Fast	Num.	Fast	Num.
-06	40	41	57	52	68	63
+00	93	91	83	86	94	94
+06	98	99	95	95	96	96
+12	100	100	100	100	95	95
+18	99	100	100	100	99	99

Table 2: Compensated Static Performance

Table 2 shows the performance of two model compensation schemes. The first, labelled *Fast*, uses the assumption that the sum of two lognormally distributed variables is itself lognormally distributed. With this assumption the compensation has a very low computational load. Secondly, numerical integration to estimate the mean and variance is used, labelled *Num.*. This should be a good approximation to the best ML estimate of the parameters given the statistics supplied. Comparing the *Fast* performance to that of *Num.* there is little difference between the two schemes. The assumption that the sum of two lognormally distributed variables is approximately lognormally distributed itself, appears from empirical results to be good. Comparing the performance of the compensation schemes to that of *Noisy* shows no significant difference. As expected the *Noisy* results are slightly better. This indicates that there is enough discriminatory information contained in single Gaussian mixture HMMs for the database used.

In order to investigate the use of delta coefficient compensation in the PMC framework, it was decided to use only the delta coefficients in the recognition. If static coefficients are incorporated they tend to dominate the recognition, having typically around 90% accuracy at $+00dB$ on this database. The first set of experiments on the delta coefficients were run assuming that the true corrupted speech variance was known. Thus only the delta coefficient means were compensated and the variance was set to the true variance of the corrupted speech. Table 3 shows the baseline performance of the clean delta coefficient means, *Clean*, and the true corrupted speech means, *Noisy*. The performance of the *Clean* parameters drops off rapidly below +12dB, whilst the *Noisy*

SNR	*Lynx*		*F16*		*Car*	
dB	Clean	Noisy	Clean	Noisy	Clean	Noisy
-06	26	29	13	22	28	33
+00	29	54	28	59	34	43
+06	66	82	77	90	68	82
+12	90	97	97	99	84	87
+18	95	97	100	100	95	98

Table 3: Baseline Delta Mean Performance

parameters achieve good performance down to +06dB.

SNR	*Lynx*		*F16*		*Car*	
dB	Fast	Num.	Fast	Num.	Fast	Num.
-06	30	29	27	25	35	33
+00	44	44	62	61	53	43
+06	82	84	88	89	79	81
+12	95	95	99	99	88	88
+18	97	97	100	100	96	97

Table 4: Compensated Delta Mean Performance

Table 4 shows the performance of the compensated means. Two compensation schemes were examined. The first used the approximation given in equation 42, labelled *Fast*. Secondly, the true ML estimate, given in equation 37, was implemented using Gaussian numerical integration. In table 4 this is labelled *Num.*. Again the variances were taken from the true variances of the corrupted speech. The *Fast* and *Num.* performance is approximately the same and comparable to the training of the parameters in noise, *Noisy*, in table 3.

SNR	*Lynx*		*F16*		*Car*	
dB	Base	Comp	Base	Comp	Base	Comp
-06	20	32	8	20	25	29
+00	27	44	28	53	37	37
+06	49	64	34	82	61	50
+12	84	80	88	96	86	71
+18	97	94	96	98	95	88

Table 5: Full Delta Compensation Performance

Table 5 shows a comparison of uncompensated delta coefficients, *Base*, with compensated delta coefficients, *Comp*. For the *Comp* scheme both means and variances were estimated using the fast approximation. The performance of both schemes are worse than if the true variance of the delta coefficients is used.

6 Conclusions

This report describes Parallel Model Combination in terms of estimating the Maximum Likelihood parameters of a corrupted speech model. The corrupted speech model is trained on statistical data obtained from the clean speech and interfering noise. These statistics are modelled by separate

HMMs. Given the statistics and the 'mismatch' function, which models the effect of the additive noise on the parameter of interest, it is simple to find expressions for the ML estimates of the corrupted model. In this report 'mismatch' functions are described for both static and delta parameters, where the speech is coded using cepstral coefficients. For the static parameters, numerical integration and a highly computationally efficient approximation are compared to training and testing the models under the same noise condition. Both the approximation and the numerical integration yield comparable recognition performance to training in the noise condition. In the case of the delta parameters, an efficient approximation for the ML estimate is derived and shown to give an estimate of the mean that yields good recognition results. However, the estimate of the variance is poor. Methods of improving this estimate are under investigation.

Acknowledgement

M. Gales is funded by a SERC studentship and a CASE award with DRA Malvern.

References

[1] Berstein AD and Shallom ID. An hypothesized wiener filtering approach to noisy speech recognition. In *Proceedings ICASSP*, pages 913–916, 1991.

[2] Varga AP, Steeneken HJM, Tomlinson M, and Jones D. The NOISEX-92 study on the effect of additive noise on automatic speech recognition. In *Technical Report, DRA Speech Research Unit*, 1992.

[3] Varga AP and Moore RK. Hidden markov model decomposition of speech and noise. In *Proceedings ICASSP*, pages 845–848, 1990.

[4] Mellor BA and Varga AP. Noise masking in the mfcc domain for the recognition of speech in background noise. In *Proceedings IOA*, volume 14, pages 503–510, 1992.

[5] Mansour D and Juang BH. The short-time modified coherence representation and noisy speech recognition. In *IEEE Trans. Acoust., Speech Signal Processing*, volume 37, pages 795–804, 1989.

[6] Klatt DH. A digital filterbank for spectral matching. In *Proceedings ICASSP*, pages 573–576, 1979.

[7] Cung HM and Normandin Y. Noise adaptation algorithms for robust speech recognition. In *Proceedings ESCA Workshop on Speech Processing in Adverse Conditions*, pages 171–174, 1992.

[8] Rabiner L.R. A tutorial on hidden Markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, volume 77, February 1989.

[9] Gales MJF and Young SJ. An improved approach to the hidden markov model decomposition of speech and noise. In *Proceedings ICASSP*, 1992.

[10] Gales MJF and Young SJ. Cepstral parameter compensation for HMM recognition in noise. *Speech Communication*, 1993. To be published.

[11] Lockwood P and Boudy J. Experiments with a non linear spectral subtractor (nss), hidden markov models and the projection, for robust speech recognition in cars. In *Proceedings Eurospeech*, 1991.

[12] Furui S. Toward robust speech recognition under adverse conditions. In *Proc. ESCA Workshop in Speech Processing in Adverse Conditions*, pages 31–42, nov 1992.

[13] Boll SF. Suppression of acoustic noise in speech using spectral subtraction. In *IEEE Transactions ASSP*, volume 27, pages 113–120, 1979.

[14] Young SJ. *HTK Version 1.4: Reference Manual and User Manual.* Cambridge University Engineering Department Speech Group, 1992.

[15] Beattie VL and Young SJ. Noisy speech recognition using hidden markov model state based filtering. In *Proceedings ICASSP*, pages 917–920, 1991.

[16] Beattie VL and Young SJ. Hidden markov model state-based cepstral noise compensation. In *Proceedings ICSLP*, pages 519–522, 1992.