State-based Cepstral Domain Compensation For Improved Noisy speech Recognition

G. Wong
Cambridge University Engineering Department,
Trumpington Street, Cambridge, CB2 1PZ, England
email: gw@uk.ac.cam.eng

July 21, 1992

Abstract

Condition mismatch in the training and testing conditions causes recognition accuracy of Hidden Markov Model (HMM) recognizers to lower substancially. An Expectation-Maximization (EM) framework is used for this problem on the set of baseline signal, mismatched signal caused by car noise and the state sequence through an N-state HMM source model. Non-iterative cepstral compensation schemes have been derived and implemented to remove the existing mismatch caused by noise. The N-vector format word modelling the baseline signal is characterized by the sample average of speech vectors in the HMM state and the average state segmentation points. The Expectation step provides the mean squared error (MSE) of the acoustic mismatch between the two types of speech signals and the calculation of the state sequence. The Maximization step consists of the calculation of N state-based compensation vectors.

State-based cepstral means compensation applied to the training material has brought good results when applied to a noisy digit database. Because the compensation is word-dependent, its application needs to be hypothesis-driven on the test material. Its application has not yielded very useful recognition results, especially for low signal-to-noise ratio noisy speech. Robustness of the method in terms of accuracy of the state segmentation boundaries and the applicability of the state-based cepstral means deviation vector at different signal-to-noise ratios is investigated. An iterative cepstral means shift technique is attempted and shown to improve (error rate reduction of 45%) on the baseline matched conditions. The implementation aspects of state-based compensation are discussed throughout with possible extensions for further improvements. Previous approaches to tackling mismatch signal modelling in speech are also described.

1 Introduction

Normally, the sensitivity of a small vocabulary speech recognizer to the absolute noise level is moderate when the system is trained and tested in the same acoustic environment. However, a mismatch between the training and testing conditions weakens the performance of standard Hidden Markov model systems. This mismatch condition arises when the recognizer does not have access to training data associated with the noise conditions in which it will be tested or the noise conditions are far too wide in range and severity that training must be effected for just a selected range of those noise conditions.

The Viterbi scoring for discrete HMMs is governed by the equation below where the 'a' and 'b' parameters have been estimated from the training environment:

$$\phi[v][j] = \max_i \Big\{ \phi[v][i] + a^v[i][j] \Big\} + b^v_j(O_t). \tag{1}$$

In an HMM, the output probability matrix connects the underlying states with the speech measurements. This represents a direct dependence on the signal-to-noise ratio (SNR) and the nature of background noise in which the recording is made. This directly affects the output probability term $b^v_j(O_t)$ where O_t is the current observation frame.

In addition, because surrounding noise affects the way somebody articulates the words (the so-called Lombard effect [1, 2]), this will have an adverse effect on the state transition matrix of a conventional HMM system trained in a different environment – the state transition matrix usually captures the different articulations of the vocal apparatus at a segmental level. For concreteness, consider some 5-state HMMs (left to right with no skips) for a digit database being obtained from clean speech and speech in a car at 100 Km/h (high-noise speech) and the average normalized state durations (table 3 and 5). The normalised state durations vary markedly in some regions between the two speech conditions e.g. for digit *one* the significant differences are over states 4 and 5, for digit *two* the differences lie in states 3 and 4 and for digit *nine* the differences are over states 2 and 3.

Conventional filtering in the spectral domain treats noisy speech on a frame-by-frame basis with the noise suppression applied directly to the magnitude FFT-spectrum [3]. Subtractive noise cepstral or spectral preprocessing generally improves SNR considerably [4], but that does not imply a proportional increase in the recognition system performance. The problem is that not all speech sounds are affected equally by subtractive-type noise reduction techniques: lower energy sounds like nasals and plosives tend to be enhanced less than the voiced portions of words with higher local SNRs. Therefore, every speech vector should not be treated independently by ascribing a probability distribution which is situated in the individual vectors of the training data because speech is clearly not an independent identically distributed (i.i.d.) source. Instead, the probability distribution of the entire training data for a particular class is parametrically modelled by a (discrete) Hidden Markov source model. In so doing, the Markov process present in an HMM is crucial because the speech vectors are highly correlated over several frames and that correlation can be modelled by the state statistics of the parametric Hidden Markov source model.

Finding a robust noise compensation technique so that the recognizer not only maintains high performance under matched conditions, but does not suffer extensive degradation when condition mismatch occurs, has become an important research thrust. By applying a simple compensation vector (cepstral means shift) to the mismatched speech vector over the entire utterance, an equalization of the condition of the reference and test material is being attempted. Part of the motivation for doing this comes from the informal observation that for many features, like the mfcc preprocessor, the major observable SNR-dependent effect is in fact a shift in the mean value for most of the components of the vector.

Later, it will be shown how by using *a priori* or iteratively determined mapping vectors over different regions of an utterance, equalization of the speech statistics of the training and testing material can be attempted. These mapping vectors are obtained by using the HMM as source model and, in particular, the property of fairly high correlation among frames attached to a particular state. The links between this approach and other proposed methods for mismatched speech recognition and, in particular, noisy speech recognition, will be delayed until section 4 – more material from the next section will provide a better perspective.

The organization of this report is as follows. Section 2 briefly reviews the Expectation-Maximization algorithm for solving Maximum Likelihood problems. In section 3, the theory underpinning cepstral means compensation to the speech material is derived using the EM framework. Various aspects of the state-based compensation scheme will be discussed along with other possible compensation schemes. Links with other related techniques are mentioned in section 4. Section 5 describes the baseline HMM set-up and recognition results. A simple cepstral compensation experiment is introduced in section 6, followed by the more detailed state-based one on the training material, but with pre-computed time boundaries in which to apply these cepstral means shift. The state-based compensation scheme with known state boundaries applied to the test material is described in section 7. Section 8 assesses the robustness of the method for the two crucial aspects of accurate segmentation breakpoints and applicability of cepstral means vector correction at various SNRs. The application of means shift with unknown state segmentation points has been attempted in an iterative way in section 9. And finally in section 10, discussions on the overall method and possible extensions of the work are presented.

2 EM theory

The principle of Expectation-Maximization (EM) theory is briefly outlined here. The main general references for this section are by Dempster and Redner [5, 6] The ones more specific to HMMs are by Baum [7, 8] and by Liporace [9].

The EM algorithm is a general approach for maximizing a likelihood or posterior (Bayesian) function when some of the data are 'missing' in some sense, and observation of that missing data would greatly simplify the estimation of parameters. Without that missing data component introduced in the likelihood function, the likelihood function of the original data may be too difficult or impractical to maximize or simply not accurate enough for some time sequential problems. In the present HMM case which models sequences of measurement vectors, data are 'missing' not because of any censoring or misrecording, but because of the superposition of the state sequence whose behaviour is governed by a first-order Markov chain. This means that the more basic data – the state from which the observation is emitted at time t cannot be observed.

Let Y be the observed 'incomplete data' which has the pdf $P(Y|\theta)$, from which we wish to estimate the parameter vector θ. The maximum likelihood (ML) estimator of θ based on the available incomplete data, is given by

$$\hat{\theta}_{ml,Y} = \arg\{\max_{\theta} L_Y(\theta)\} = \arg\{\max_{\theta} \log P(Y|\theta)\}, \tag{2}$$

where $L_Y(\theta)$ is the likelihood function of the incomplete data.

At this point, it is assumed that the complete data X have been chosen in such a way that computing the ML estimator of θ from the complete data, i.e. solving

$$\hat{\theta}_{ml,X} = \arg\{\max_{\theta} L_X(\theta)\} = \arg\{\max_{\theta} \log P(X|\theta)\} \tag{3}$$

is significantly simpler than solving (2). $L_X(\theta)$ is the likelihood function of the complete data.

The incomplete data is related to the complete data X through a non-invertible many-to-one transformation:

$$Y = J(X). \tag{4}$$

The transformation $J(.)$ relating X to Y can be any non-invertible transformation. There may be many possible complete data specifications that will generate the observed data – the EM algorithm can therefore be implemented in many possible ways. The formulation of the complete data is crucial because a good one will reduce the complexity and convergence time of the algorithm. The pdf of X, which is also indexed by θ, is related to Y as follows:

$$P(Y|\theta) = \int_{X(Y)} P(X|\theta)\, dX. \tag{5}$$

The probability distribution of the parameter vector θ conditioned on the data vector Y using Bayes rule is:

$$P(\theta|Y) = \frac{P(Y|\theta)P(\theta)}{P(Y)}. \tag{6}$$

For Bayesian estimation, some *a priori* information is incorporated in the estimate by specification of the prior $P(\theta)$. Since $P(Y)$ is a constant for a given data vector belonging to a particular class, Maximum *a posteriori* (MAP) estimation yields the following statement:

$$\max_{\theta} B(\theta|Y) = \log P(Y|\theta) + \log P(\theta). \tag{7}$$

The term $B(\theta|Y)$ is known as the Bayesian or log posterior function.

It then follows that due to this many-to-one X to Y mapping

$$P(X|Y;\theta) = \frac{P(X,Y|\theta)}{P(Y|\theta)} = \frac{P(X|\theta)I_Y(X)}{P(Y|\theta)}, \tag{8}$$

where $I_Y(X)$ is the indicator function which is equal to 1 if X results in Y and equal to 0 otherwise. For an HMM, this indicator is the state from which the observation is emitted.

Consider the denominator term of (8). For any $\hat{\theta}^n$, the parameter vector at the n-th iteration in the reestimation algorithm, the following relationship applies:

$$E_X\Big\{\log P(Y|\theta)|Y;\hat{\theta}^n\Big\} = \int_{X(Y)} \log P(Y|\theta)P(X|Y;\hat{\theta}^n)dX = \log P(Y|\theta), \tag{9}$$

where $E_X\{.\}$ is the expectation operator with respect to X, the complete data. $P(X|Y;\hat{\theta}^n)$ is the conditional pdf of the complete data, given the incomplete data and the estimate $\hat{\theta}^n$.

Combining (8) and (9) by taking log and then the expectation operator gives

$$\log P(Y|\theta) = E_X\Big\{\log P(X|\theta)|Y;\hat{\theta}^n\Big\} - E_X\Big\{\log P(X|Y,\theta)|Y;\hat{\theta}^n\Big\}. \tag{10}$$

Substituting (10) into (7) results in

$$B(\theta|Y) = Q(\theta|\hat{\theta}^n) - E_X\Big\{\log P(X|Y,\theta)|Y;\hat{\theta}^n\Big\}, \tag{11}$$

where

$$Q(\theta|\hat{\theta}^n) = E_X\Big\{\log P(X|\theta)|Y;\hat{\theta}^n\Big\} + \log P(\theta). \tag{12}$$

The first term is the conditional expectation of the log likelihood of the complete data, given the observed incomplete data Y and $\hat{\theta}^n$.

So, (12) can be re-written as:

$$Q(\theta|\hat{\theta}^n) = L(\theta|\hat{\theta}^n) + \log P(\theta), \tag{13}$$

where

$$L(\theta|\hat{\theta}^n) = E_X\Big\{\log P(X|\theta)|Y;\hat{\theta}^n\Big\} \tag{14}$$

From Jensen's inequality [10] (pages 49-50) for any $\hat{\theta}^{n+1} \neq \hat{\theta}^n$,

$$E_X\Big\{\log P(X|Y,\hat{\theta}^{n+1})|Y;\hat{\theta}^n\Big\}\leq E_X\Big\{\log P(X|Y,\hat{\theta}^n)|Y;\hat{\theta}^n\Big\}, \tag{15}$$

with equality if and only if

$$\log P(X|Y;\hat{\theta}^{n+1}) = \log P(X|Y;\hat{\theta}^n). \tag{16}$$

It follows that a sufficient condition for $B(\hat{\theta}^{n+1}|Y) > B(\hat{\theta}^n|Y)$ is $Q(\hat{\theta}^{n+1}|\hat{\theta}^n) > Q(\hat{\theta}^n|\hat{\theta}^n)$ since the second term of (11) is guaranteed not to decrease by Jensen's inequality.

In general, if $B(\theta|Y)$ is not unimodal, the EM approach at best assures convergence of the sequence $\{\boldsymbol{\theta}^{\mathbf{n}}\}$ to a stationary value. The convergence point will normally not be the global maximum of the object function for a complicated problem – several starting points in the initial parameter vector space may be needed to locate the best maxima. For the finer points of the convergence of the EM algorithm, refer to Wu's work [11].

The EM algorithm is the application of those two steps below, in an iterative way, until a pre-defined threshold is attained.

The E-step

Starting with an estimate of the parameter vector $\hat{\theta}^n$ for the n-th iteration of the reestimation algorithm, the EM algorithm for MAP estimation consists of the Expectation step (E-step) whereby

$$E_X\Big\{\log P(X|Y,\theta)|Y;\hat{\theta}^n\Big\} \tag{17}$$

is formed i.e. the expected value of the logarithm of the pdf of the complete data is evaluated, where the expectation is with respect to the probability measure defined by the incomplete data and the current parameters. The E-step thus finds the conditional expectation of the sufficient statistic for the complete data log likelihood. Note that the E-step is not affected by the prior term.

The M-step

The Maximization step (M-step) corresponds to the maximization of the log likelihood function, $L(\theta|\hat{\theta}^n)$ with respect to θ. This leads to a new parameter estimate $\hat{\theta}^{(n+1)}$:

$$\hat{\theta}^{(n+1)} = \arg\{\max_{\theta} L(\theta|\hat{\theta}^{(n)})\}. \tag{18}$$

For the posterior function, solve

$$\arg\{\max_{\theta} Q(\theta|\hat{\theta}^n)\}, \tag{19}$$

where

$$Q(\theta|\hat{\theta}^n) = E_X\Big\{\log P(X|\theta)|Y;\hat{\theta}^n\Big\}+\log P(\theta). \tag{20}$$

The complete data log likelihood is maximized with respect to the unknown parameters, with the conditional expectation of the sufficient statistics substituted in place of their unknown values. The choice of the prior function will affect this maximization step of the EM algorithm for the posterior function.

For the log likelihood function where $L_X(\theta)$ is defined on the true complete data, $L(\theta|\hat{\theta}^{(n)})$ uses the conditional expectation of the complete data. The maximization of $L(\theta|\hat{\theta}^{(n)})$ with respect to θ is therefore of the same complexity as the maximization of $L_X(\theta)$. Because of this, the EM algorithm is an attractive alternative to the direct evaluation of (2) only if the solution to (3) can be computed relatively easily. Solving the more general Q function with the prior term depends on a judicious choice of this prior term. If components of the complete data X are independent, the complete data likelihood function is a linear function of the incomplete data – the M-step only requires the optimization of a set of those functions. This decoupling is fundamental to signal mismatch problems, as will be seen in the subsequent sections.

3 EM treatment

The statement of the problem is as follows. Given some speech samples of a baseline signal and some speech samples of a mismatched signal (from the same class) where the mismatch is caused by noise, how are the statistics of the mismatched signal to be modified in such a way that the resulting statistical characteristics come as close as possible to the baseline signal (in effect the training data)? On top of that, when carrying out the transformation, how is the average timing information kept, in the sense that the sequence of low and high-SNR regions in the two different speech conditions ought to correspond? The first question is answered by using a simple cepstral means compensation vector applied to the mismatched signal. The second question is answered by using the correspondence afforded by HMMs that have been aligned to the two different speech conditions and the resulting HMM state-to-state alignment. The EM framework will be used for this mismatch and, in particular, the noisy speech recognition problem.

The techniques developed are expected to apply to cases such as different telephone sets and microphones which result in acoustic mismatches between the gathered speech data, different background noise conditions and room acoustics, and mismatches incurred in speech recordings from different voice transmission systems. The mismatch can also arise from different speakers' utterances, stressed speech or different speakers' codebooks. Although discrete HMMs are used, most of the compensation techniques developed extend to continuous density HMMs in a straighforward way, simply because the modification is applied to the speech vectors before the intervening HMM training or recognition phase.

A speaker-dependent, digit vocabulary database will be the particular focus of the compensation process. Two assumptions will be made about the nature and effect of the noise disturbance on the speech that will subsequently yield a simplified compensation scheme. Firstly, only slowly changing noise at a spectral level (car noise can be considered to fit into this category) will be considered. Secondly, a nominal SNR treatment will be followed. The SNR dependency will be global throughout all the experiments in that only the nominal SNR of the speech needs to be known, despite the widely varying instantaneous SNR of the individual speech frames.

Section 3.1 introduces the *incomplete* and *complete* data set for the mismatch problem, the relationship between the *incomplete* and *complete* data and the parameter vector. The E-step and M-step are applied to the different signals in the cepstral domain to derive state-based compensation vectors, the state sequence and the length-normalized mean squared error (MSE) of the two speech conditions. The overall effect of the compensation is to bring about a length-normalized MSE reduction between the two types of speech conditions. Section 3.2 discusses some aspects of the state-based compensation scheme. Finally, by dropping and/or relaxing the state-based mapping between the two different speech signals, other compensation schemes are possible and this is described in section 3.3.

3.1 EM formulation

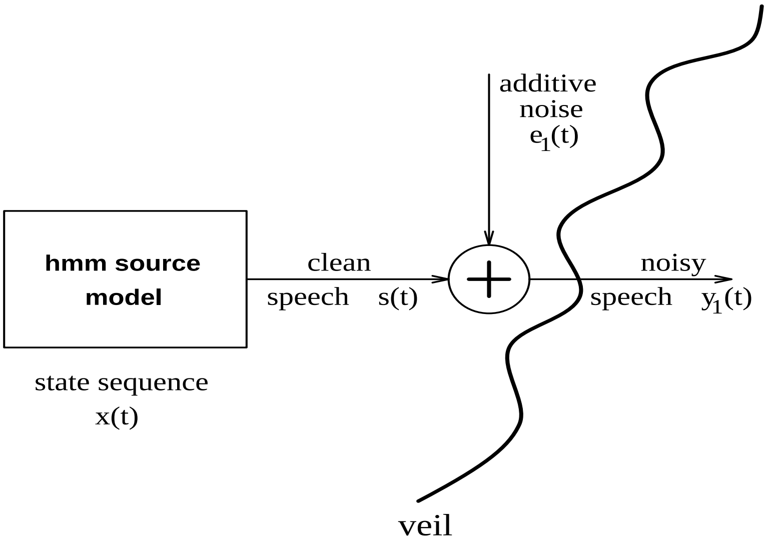


Figure 1: Source and noise model used in mismatched speech recognition.

$e_1(t)$ can be regarded as the mismatch statistics between the input signal $y_1(t)$ and the signal from the Hidden Markov source model $s(t)$ with state sequence $x(t)$. The *incomplete* but *observable* data Y is $y_1(t)$. The *complete* data X are postulated to be the set of signals $\{s(t), x(t), y_1(t)\}$. In Ephraim's work [12], in effect, a slightly different *complete* data set is used ($s(t)$, $x(t)$, $y_1(t)$ and $h(t)$, the mixture component from a continuous density HMM). Dembo used a slightly different *complete* data set in analyzing the signal reconstruction of data from noisy transforms [13]. The interactions between the different signals are shown diagrammatically in figure 1 and the notations are listed below.

N	total number of states in a Markov chain
T	total number of frames in the utterance
$y_1(t)$	input speech mfcc vector
$s(t)$	corresponding baseline speech mfcc vector
$x(t)$	state sequence through the Markov chain of the HMM ($\in [1, N]$) modelling the baseline signal $\mathbf{s}$
$e_1(t)$	corresponding 'noise' mfcc vector

$y_1(t)$ is a vector but will not be denoted in bold. Instead, the whole sequence $\{y_1(t)\}$ from $t = 1$ to $t = T$ will be denoted by the bold font term $\mathbf{y}_1$. The same applies to the sequences $x(t)$, $e_1(t)$ and $s(t)$. The unknown parameter vector θ is the sequence of cepstral means deviation vectors $\{e_1(t)\}$. Assume, for the moment, that $\mathbf{y}_1$ and $\mathbf{s}$ belong to the same class. The following additive signal and noise model used in figure 1 has the relationship:

$$y_1(t) = s(t) + e_1(t) \tag{21}$$

Suppose that the complete data are available. The corresponding complete-data likelihood is:

$$\begin{aligned} L_c(\theta) &= \log f_{\mathbf{y}_1,\mathbf{x},\mathbf{s}}(\mathbf{y}_1, \mathbf{x}, \mathbf{s}; \theta) && \text{(22a)} \\ &= \log f_{\mathbf{y}_1|\mathbf{s},\mathbf{x}}(\mathbf{y}_1|\mathbf{s}, \mathbf{x}; \theta) + \log f_{\mathbf{s},\mathbf{x}}(\mathbf{s}, \mathbf{x}; \theta) && \text{(22b)} \\ &= \log f_{\mathbf{s}|\mathbf{y}_1,\mathbf{x}}(\mathbf{s}|\mathbf{y}_1, \mathbf{x}; \theta) + \log f_{\mathbf{s},\mathbf{x}}(\mathbf{s}, \mathbf{x}; \theta) + K, && \text{(22c)} \end{aligned}$$

where K is a constant term likelihood. Bayes' rule is used in the last step, i.e.

$$f(\mathbf{y}_1|\mathbf{s}, \mathbf{x}) = \frac{f(\mathbf{y}_1)f(\mathbf{s}|\mathbf{y}_1, \mathbf{x})}{f(\mathbf{s})}. \tag{23}$$

Note the decoupling of the Hidden Markov source model term and the signal term.

The E-step

By using the linear property of the Expectation operator,

$$E(A + B) = E(A) + E(B) \tag{24}$$

and from (22), the E-step involves the calculation:

$$\begin{aligned} E_X\Big\{\log P(X|\theta)|Y; \hat{\theta}^n\Big\} &= E_{\mathbf{s}|\mathbf{y}_1,\mathbf{x}}\Big\{\log f_{\mathbf{s}|\mathbf{y}_1,\mathbf{x}}(\mathbf{s}|\mathbf{y}_1, \mathbf{x}, \theta)|\mathbf{y}_1, \hat{\theta}^n\Big\} && \text{(25a)} \\ &\quad + E_{\mathbf{s},\mathbf{x}}\Big\{\log f_{\mathbf{s},\mathbf{x}}(\mathbf{s}, \mathbf{x}|\theta)|\mathbf{s}, \hat{\theta}^n\Big\}. && \text{(25b)} \end{aligned}$$

A Viterbi decoding is first carried out to obtain the most likely state sequence (θ^n are the parameter estimates at the n-th iteration). This is one of the key aspects of the E-step in order to take advantage of the correlated property of frames attached to states in the HMM. Thus,

$$\hat{\mathbf{x}} = \mathbf{x}^\star = E[\mathbf{x}|\mathbf{y}_1; \theta^n]. \tag{26}$$

The second term in (25) can be simplified as follows:

$$\begin{aligned} E_{\mathbf{s},\mathbf{x}^\star}\Big\{\log f_{\mathbf{s},\mathbf{x}^\star}(\mathbf{s}, \mathbf{x}^\star|\theta)|\mathbf{s}, \hat{\theta}^n\Big\} &= E_{\mathbf{s},\mathbf{x}^\star}\Big\{\log f(\mathbf{x}^\star; \hat{\theta}^n) f(\mathbf{s}|\mathbf{x}^\star; \hat{\theta}^n)\Big\} && \text{(27a)} \\ &= \prod_{t=1}^{T} \log a_{x^\star_{t-1} x^\star_t} b_{x^\star_t}(s_t) && \text{(27b)} \end{aligned}$$

with the 'a'' and 'b'' elements being those of the HMM modelling the baseline signal $\mathbf{s}$. The initial and final state transition matrix can be ignored for the current purpose.

Thus, (25) can now be re-written as:

$$\begin{aligned} E_X\Big\{\log P(X|\theta)|Y; \hat{\theta}^n\Big\} &= E_{\mathbf{s}|\mathbf{y}_1,\mathbf{x}^\star}\Big\{\log f_{\mathbf{s}|\mathbf{y}_1,\mathbf{x}^\star}(\mathbf{s}|\mathbf{y}_1, \mathbf{x}^\star, \theta)|\mathbf{y}_1, \hat{\theta}^n\Big\} && \text{(28a)} \\ &\quad + \prod_{t=1}^{T} \log a_{x^\star_{t-1} x^\star_t} b_{x^\star_t}(s_t) && \text{(28b)} \end{aligned}$$

The first term of (28) is the mean squared error estimate of the acoustic mismatch between the baseline speech $s(t)$ and the observed speech $y_1(t)$ [14]. The second term in (28) is exactly the HMM E-step but with a single dominant sequence through the Markov chain and has already been shown to simplify to the calculation of the log likelihood of the observation [15].

The M step

The matching criterion between the two types of speech $\mathbf{y}_1$ and $\mathbf{s}$ is chosen to be the mean squared error because of its mathematical tractability. The noise vector sequence $\mathbf{e}_1$ is postulated to be the sequence of state-based vectors $\{\mathbf{c}_p, 1 \leq p \leq N\}$, which is applied to the input mismatched signal.

The update of the signal $e_1(t)$ is given (from (18)) by:

$$\begin{aligned}
\hat{\theta}_{\mathbf{e}_1}^{(n+1)} &= \arg\Big[\max_{\theta} E_X\Big\{\log P(X|\theta)|Y;\hat{\theta}^n\Big\}\Big] && \text{(29a)}\\
&= \arg\Big[\max_{\mathbf{e}_1}\Big\{\log f_{\mathbf{s}|\mathbf{y}_1,\mathbf{x}}(\mathbf{s}|\mathbf{y}_1,\mathbf{x}^\star,\theta)|\mathbf{y}_1,\mathbf{x}^\star,\hat{\theta}^n\Big\} && \text{(29b)}\\
&\quad + \max_{\mathbf{e}_1}\Big\{\prod_{t=1}^{T}\log a_{x_{t-1}^\star x_t^\star} b_{x_t^\star}(s_t)\Big\}\Big] && \text{(29c)}\\
&= \arg\Big[\min_{\mathbf{e}_1}\Big\{||\mathbf{y}_1 - \mathbf{s}|| \mid \mathbf{y}_1,\mathbf{x}^\star,\hat{\theta}^n\Big\} + \max_{\mathbf{e}_1}\Big\{\prod_{t=1}^{T}\log a_{x_{t-1}^\star x_t^\star} b_{x_t^\star}(s_t)\Big\}\Big] && \text{(29d)}\\
&= \arg\Big[\mathrm{mse}(\mathbf{x}^\star)\Big], && \text{(29e)}
\end{aligned}$$

where

$$\mathrm{mse}(\mathbf{x}^\star) = \min_{\mathbf{c}_p}\Big\{\sum_{p=1}^{N}\frac{1}{T[p]}\sum_{t=1}^{T[p]} \parallel (y_1(t)_{\mathbf{x}^\star} - \mathbf{c}_p) - s(t)_{\mathbf{x}^\star} \parallel^2 \Big\} + \prod_{t=1}^{T}\log a_{x_{t-1}^\star x_t^\star} b_{x_t^\star}(s_t). \qquad (30)$$

$T[p]$ is the number of frames for which p is the HMM state index among all the frames in the speech material concerned.

By differentiating with respect to $\mathbf{c}_p$, the final state-based correction vectors can easily be shown to be:

$$\mathbf{c}_p = \frac{1}{T[p]}\sum_{t=1}^{T[p]} y_1(t)_{\mathbf{x}^\star} - \frac{1}{T[p]}\sum_{t=1}^{T[p]} s(t)_{\mathbf{x}^\star} \qquad 1 \leq p \leq N. \qquad (31)$$

(31) applies for a single utterance but is easily generalized to multiple utterances since these are independent of each other. Compensation vectors can be applied to the baseline signal $s(t)$ instead of the incoming signal $y_1(t)$ and similar compensation vectors to (31) can be derived with the signs reversed. Alternative ways of providing correspondence between the mismatched and baseline signals will be explored in subsection 3.3. From (28), $\mathrm{mse}(\mathbf{x}^\star)$ (refer to (30)) with $\mathbf{c}_p$ plugged in represents the actual E-step and partly corresponds to the length-normalized MSE between $\mathbf{y}_1$ and $\mathbf{s}$ – this allows the iterative process to be monitored. Combining the scores of the acoustic mismatch and the log likelihood is possible but depends on a bias term to give equal weightings to the two terms. This avenue has not been investigated in this dissertation.

(31) implies that $s(t)$ has to be aligned by $\mathbf{x}^\star$ derived from the mismatched signal $\mathbf{y}_1$. But normally many instances of the training material are present, so a more accurate segmentation

can be derived, namely

$$\mathbf{x}^{\text{train}} = E[\mathbf{x}|\mathbf{s};\theta^n]. \tag{32}$$

Thus, (31) is modified into:

$$\mathbf{c}_p = \frac{1}{T[p]}\sum_{t=1}^{T[p]} y_1(t)_{\mathbf{x}^\star} - \frac{1}{T[p]}\sum_{t=1}^{T[p]} s(t)_{\mathbf{x}^{\text{train}}} \qquad 1 \le p \le N. \tag{33}$$

If many utterances characterising the speech signal $\mathbf{y}_1$ are available and the identity of the class is known, $\mathbf{x}^\star$ can be computed from these samples. This allows the calculation of the average state sequence (or the average normalized state segmentation points) and the average state statistics for both signals $\mathbf{y}_1$ and $\mathbf{s}$ (the training data). Thus, non-iterative compensation schemes can be derived because of the *a priori* determined state-based compensation vectors from (33).

If previous samples of signal $\mathbf{y}_1$ are not available, an iterative compensation scheme is required. The E-step consists of

1. The estimation of the MSE of the mismatch between the two speech signals i.e. (30) with vectors calculated by (33) from the previous iteration plugged in. The bias introduced by the HMM term is disregarded.

2. The calculation of the state sequence, or equivalently the state segmentation boundaries, given the estimated cleaner speech signal $\hat{y}_1(t)$ from the previous iteration i.e. (26).

The M-step consists of

1. The calculation of the state-based compensation vectors i.e. (33) with the state sequence derived in the E-step.

By alternating between the E-step and the M-step, the statistics of the incoming signal are brought closer to the statistics of the baseline reference signal. Better recognition results ought to follow.

To summarize, the incomplete data are $Y = \{y_1(t)\}$, the complete data are $X = \{y_1(t), s(t), x(t)\}$ and the parameters to be estimated $\theta = \{e_1(t)\}$, $1 \le t \le T$. The relationship between X and Y is through these two elements:

1. The baseline signal $s(t)$ is related to the HMM state sequence $x(t)$. This link has already been discussed by Wong [15].

2. An additive signal and noise model is used:

$$y_1(t) = s(t) + e_1(t). \tag{34}$$

The E-step and M-step have been derived in the discussion above.

Implementation details of the collection of state-based compensation vectors necessary for non-iterative compensations schemes will be given in subsection 6.2 after some baseline experiments (section 5) that have been carried out are described. The next section (3.2) describes some aspects and implications of using a state-based compensation scheme in tackling mismatch speech recognition caused by noise.

3.2 State-based compensation scheme

The operation consisting of the use of a word-dependent and state-dependent cepstral means shift vector in compensating for the mismatched speech condition will henceforth be called state-based cepstral (means) domain compensation (SBCDC). At the mel-scale cepstral vector level, the following transformation is applied at a particular SNR:

$$(\hat{y}_1)_t = s_t + \mathbf{c}_p \qquad 1 \leq p \leq N \tag{35}$$

where $(\hat{y}_1)_t$ is the compensated mfcc vector, s_t is the 'clean' mfcc vector, all at time t, and $\mathbf{c}_p$ [1] is the state-based compensation vector. This implies that armed with some knowledge of the compensation vector a simple vector subtraction is enough to recover the clean speech vector at each frame. Conversely, the noise disturbance vector can be added to some 'clean' vector to eventually give a sequence of vectors whose distribution and overall statistical characteristics correspond to that of the noisy speech.

Because of the state dependency, (35) is assumed valid over a partial region of the speech associated with the frames in between the segmentation points achieved by an HMM in dividing the speech into quasi-stationary regions. This should be contrasted to the work by Chen [16] whereby the so-called additive stress component is constant throughout the whole utterance. The first-order Markov property of the HMM state transition process and the correlated frames it encompasses, even for noisy signals, makes the speech estimates in adjacent frames state-dependent. The validity of this approach is further made more realistic by consideration of these two limits: often, up to 3-4 frames across parts of the vector quantization (VQ) encoded speech for speaker-dependent digit vocabulary and fairly low-order codebook are identical and for a larger portion of speech covered by a phoneme the associated speech vectors might be affected equally by the disturbance vector.

The compensation vector $\mathbf{c}_p$ can be considered to depend on the speech sound concerned. It is reasonable to hypothesize that fricatives, for example, will be affected differently in noise at the spectral/cepstral level than say, vowels. Generally speaking, this implies that the different phonemes comprising the different digits are affected differently by (car) noise. Certainly, work by Stanton and Pisoni [1, 2] shows different patterns of energy migration in the frequency domain for English phonemes in clean and Lombard conditions. A similar conclusion can be postulated to hold for noisy speech as the Lombard effect is inherent in the process. Because the dependency implies some subword detection, and the different digits have sufficiently distinctive sequences of phonemes, this disturbance vector can be assumed to be word-dependent. More specificity will be present in using these statistics later on.

In the EM framework, the source model generating the clean speech signal is an HMM, and it seems natural for SBCDC to directly use HMM B probabilities and a knowledge of the codevectors to derive a compensation vector rather than the sample average of the terms present in (33). However, this is possibly ill-advised, and shown experimentally to be so later on, even when the original derivation requires the HMM as source model. The argument runs as follows: Because each HMM state represents distributions of feature vectors over several frames, the training data speech vectors belonging to state i of the reference model in SBCDC

[1] Depending on the application, $\mathbf{c}_p$ without the state dependency has variously been known as compensation, correction, adaptation, equalization, deviation, shift, reconstruction, adjustment, restoration, restitution, normalization, mapping or 'stress component' vector in the various noisy speech, speech enhancement, spectral mapping, stress compensation, codebook adaptation and speaker adaptation literature.

could be averaged by:

$$\text{vector}_i = \sum_{j=1}^{M} b_{ij} V_j, \tag{36}$$

where b_{ij} is the probability of the j-th codeword emitted from state i, V_j is the j-th codeword and M is the number of codevectors. But that HMM is normally trained by a total likelihood criterion, which is a sum of probabilities over all possible state sequences. In contrast, the 'best' state sequence imposed in the E-step derivation of SBCDC allows for an approximate but simplified cepstral compensation scheme. The discrepancy between these two ways of traversing the Markov chain may explain why representing the average vector in a state by the above summation consistently gives marginally lower recognition results (section 7).

In addition, errors arising from VQ encoding and inadequately trained or incorrectly modelled B parameters make it preferable to pick vectors pertaining to a state from the actual training vectors and do the averaging thereafter. Besides, the parametric Hidden Markov source model is invoked in the derivation only to simplify the systematic calculation of correlated quasi-stationary regions of speech and their boundaries.

The probability distributions of the speech baseline signal and the input mismatched signal statistics will first be estimated from long training sequences from the two processes by estimating average vectors over some pre-calculated regions (section 6) or regions subsequently found in an iterative process (section 9). Then, estimation of the attempted closer matched speech signal is obtained by doing a means shift in the corresponding mapped regions of the two speech conditions. Correlation among frames belonging to an HMM state and the average timing information is preserved in this way. The investigation and specific use of the means shift directed on the training data is described in subsection 6.3, on the test data in section 7 and on both train and test data simultaneously in subsection 7.5. Robustness of these pre-calculated state segmentation points is dealt with in section 8. Applicability of the cepstral means information in the pre-calculated regions to speech with different SNRs is investigated in the same section (8).

3.3 Discussions of other compensation schemes

Other possible correspondences between the baseline and mismatched signal are possible. The state dependency can be dropped, in which case the equivalent normalised MSE (30) is modified to:

$$\text{mse} = \min_{\mathbf{c}^0} \Big\{ \frac{1}{T} \sum_{t=1}^{T} \| \, (y_1(t) - \mathbf{c}^0) - s(t) \, \|^2 \Big\}. \tag{37}$$

By differentiating with respect to $\mathbf{c}^0$, the final correction vectors can easily be shown to be:

$$\mathbf{c}^0 = \frac{1}{T} \sum_{t=1}^{T} y_1(t) - \frac{1}{T} \sum_{t=1}^{T} s(t). \tag{38}$$

This word-based, whole-utterance compensation thus applies an average correction to the sequence of input vectors. Normally, the sample average of the reference or noisy vectors are obtained from many utterances. Experimental results using this simple technique are given in subsection 6.1. However, the piecewise treatment of SBCDC to reduce the mean squared error will be shown in subsection 6.3 to result in better recognition results compared to the simple treatment of a constant vector over the entire utterance.

This algorithm has been undertaken for mismatched training and testing microphones where a fixed additive correction vector is applied to the cepstral coefficients [17]. The compensation vectors are estimated with a minimum mean squared error criterion by computing the average difference between cepstral vectors for the test condition versus the standard acoustic training environment. This technique provides a considerable improvement when the system is trained and tested on different microphones.

The state-based mapping can also be relaxed in the following way. A VQ-based mapping has been postulated by Feng for a complex modelling of the two cepstral spaces [18]. However, a warping function $\psi \equiv (i(t), j(t))$ needs to be defined now between the two speech signals by using a Dynamic Time Warping algorithm. In effect, the reference (or mismatched) vector sequences can be shifted by a set of VQ-dependent vectors $\{\mathbf{c}_1, \cdots, \mathbf{c}_y, \cdots, \mathbf{c}_Y\}$ where Y is the total number of codevectors present in the codebook. Let $T[y]$ be the total number of frames for which y is the VQ index among all the frames in the speech material concerned. Analogous to (30), the normalised MSE term will be:

$$\mathrm{mse}(\psi) = \min_{\mathbf{c}^y} \Big\{ \sum_{y=1}^{Y} \frac{1}{T[y]} \sum_{t=1}^{T[y]} \| (y_1(t)^y_{i(t)} - \mathbf{c}^y) - s(t)^y_{j(t)} \|^2 \Big\}. \tag{39}$$

By differentiating with respect to $\mathbf{c}^y$, the final correction vectors can easily be shown to be:

$$\mathbf{c}^y = \frac{1}{T[y]} \sum_{t=1}^{T[y]} y_1(t)^y_{i(t)} - \frac{1}{T[y]} \sum_{t=1}^{T[y]} s(t)^y_{j(t)} \qquad 1 \leq y \leq Y. \tag{40}$$

The timing of the mismatch statistics is captured by the rate of variation of the VQ indices. The correction vector will be different for every codebook vector.

4 Links with other methods

Some of the works described briefly below attempt to find a probabilistic, empirical or deterministic method for mapping the mismatched speech vectors caused by stressed conditions, environment noise, different speakers or different microphones into the space of the baseline material. For example, if noise is the main cause of the mismatch, the mapped estimates are then used to correct back a given noisy speech sequence into a cleaner one.

No attempt is made in this work to divide the utterances into their phonemic constituents, unlike Bocchieri's work [19] in which an explicit time template of the acoustic events characterizing a particular word has to be pre-defined. The cepstral changes, as monitored by the HMM states, guide the selection of marked acoustic events rather than their relationship with the traditional, but more linguistically relevant phonemic context. On the other hand, the utterance could be divided linearly into 5 sections as in Shore's work [20], but the segmental properties associated with HMM state breakpoints will be missed.

A correspondence between the clean and the noisy signal can be established through spectral mapping [21]. Here, in effect, a VQ-dependent mapping is provided, and the mapping is a deterministic one. The inverse mapping produces the restored spectrum. The approach by Nadas [22] is to provide a piecewise mapping which is encoded by a linear model. The model parameters change in the two different speech conditions from one fairly stationary region to another. In SBCDC, the quasi-stationary region is covered or presumed to be covered by an

HMM state and the mapping from the two different speech conditions is just a means shift in the preprocessing parameters.

The work by Gish merits special attention because the state dependency and the amount of correlation that exists among nearby frames are not put to use explicitly [23]. Instead, a VQ-dependency and SNR-dependency is presupposed with a more general mapping in the form of a linear model. The relation between the clean and noisy vectors was defined as:

$$x_c = A(k,\gamma_j)x_n + b(k,\gamma_j) + \epsilon, \tag{41}$$

where x_c and x_n are the clean and noisy vectors respectively, ϵ is the zero mean Gaussian vector, γ_j means that the SNR is in the j-th quantization region and k refers to the VQ bin. The same transformation is used in Roucos's work [24] to improve the match between the reference and the input speakers for vocoding purposes.

Acero attempts a probabilistic mapping as well, although now the VQ-dependency is dropped in favour of all the codewords supplying an amount of information to correct the mismatched speech vectors [17]. In Porter's work [14], an empirical mapping is carried out – a clean speech database and a simulated noisy version is used to construct a function that maps a noisy spectral component at each single frequency to a noise-suppressed value. Using the ratio of sample averages to approximate the optimal estimator has the significant practical advantage that the *a priori* distribution of the uncorrupted speech vectors need not be known.

Explicit noise compensation is possible. Noisy speech vectors can be modified from *a priori* estimated noise and speech signal pdfs and their cumulative pdfs by the key assumption that at each time frame either the speech or noise is the dominant signal. The MIXMAX model [25] involves creation of noise-compensated pdfs which are then used to facilitate VQ labelling of each noisy spectral vector. Essentially, the same task is attempted by Varga [26]. The problem is to obtain the best estimate of the likelihood of an input observation conditioned on a particular HMM state which models the clean speech, and given an estimate of the amount of noise present from a noise estimator.

The objective of using compensation techniques in speaker adaptation is to establish mapping rules for each test speaker which will change the test utterance spectra to resemble the reference set spectra. In codebook adaptation, the test utterance represented by a string of VQ indices remain untouched whereas the codebook which clusters the reference speech is modified. In Roe's work [27], the spectral shape of each codevector is modified to simulate the Lombard effect and an estimate of the background noise is added to each codevector. The work by Furui attempts to hierarchically modify the codebook for a new speaker by a series of mean shifts [28]. The key aspect in his procedure was to keep the continuity of the adjacent clusters in order to preserve the phonetic information as much as possible and to keep some timing information as well.

The major effect of stress conditions (like being angry, talking loudly, etc.) is considered to be a spectral tilt relative to normal speech, and so by capturing the amount of tilt, the original normal spectrum can be restored. Multi-style training explicitly uses the stressed speech to create HMMs [29] . Chen assumes that the stress component modelling the loud or shout condition of speech remains unchanged within the word interval [16]. A slightly modified relationship to (35) is used:

$$\mathbf{y}_t = \mathbf{x}_t + \mathbf{n}_t + \delta_t, \tag{42}$$

where $\mathbf{y}_t$ is the noisy vector, $\mathbf{x}_t$ is the 'clean' vector, $\mathbf{n}_t$ is the noise vector (called the stress component) and the last term δ_t simulates the randomness of the clean speech parameters.

The stress component is derived from the sample average of the observation sequence and from a weighted sample average governed by state occupancy over all the states of a hypothesised HMM reference word. Explicit formant compensation has been carried out by Hansen [30].

There are several points of similarity between SBCDC and the speech enhancement scheme by Ephraim [12]. The scheme centers on using the correlated property of frames associated with an HMM state to apply a set of Wiener filters to correct the corresponding noisy speech frames. The filtering of a noisy region of speech depends on the decoded state sequence of a CDHMM which models all the vocabulary set at once. The particular state (and mixture component) dictates the choice of a particular Wiener filter out of a finite set which has been trained on clean speech. The filtering depends as well on the knowledge of the noise process which has been modelled separately, whereas in SBCDC the 'noise' component is modelled empirically on a digit-dependent and region-dependent (covered by an HMM state) vector, which is an estimated sample average rather than from an assumed parametric distribution. Ephraim uses a Wiener filter to enhance the noisy sequence whereas SBCDC attempts the equalization of the two speech conditions in the cepstral domain.

The region over which to apply Wiener filtering or SBCDC can be determined *a priori* given some samples of the clean speech or iteratively given that the only bootstrap is the mismatched signal in the first place. However, because of the word-dependent nature of the compensation in SBCDC, application to the test material is more problematic. Moreover, speech recognition objectives are different from those of speech enhancement. As long as the pattern and level of distortion introduced by the several stages from preprocessing, VQ and Hidden Markov modelling inaccuracies are the same during training and testing of the recognizer, discrimination can be left largely unaffected.

5 Baseline set-up

This section describes briefly the whole set-up for conventional digit HMM recognition. This includes preprocessing, vector quantization, HMM training and recognition, and the baseline results obtained.

5.1 Speech database and Hidden Markov Model details

In the absence of widely available noisy databases, a locally recorded one was used. The database contains ten digits, recorded by one speaker in three different noise environments: 36 examples of each digit in a noise-free environment (referred to as clean speech), 18 examples in a stationary car with the engine running (referred to as low-noise speech) and 18 examples in a car being driven at nominally 100 Km/h (referred to as high-noise speech) and at an estimated SNR of -7 dB. The sampling rate is 16 KHz and the speech material has been hand-segmented.

A single mel-frequency scale cpestral coefficient (mfcc) codebook was used [31]. Unlike in phoneme and word recognition tasks, inclusion of the unmodified intensity value in each vector or as an independent preprocessor may reduce noisy speech recognition performance as, for instance, demonstrated by Noll [32]. The inclusion of differential mfcc coefficients through another codebook or affixed to the current vector will generally reinforce recognition rate. However, for simplicity, this inclusion has not been put into effect.

The variations in speech mel-frequency scale cepstral coefficients of clean and noisy speech have first been studied. When examining the distribution of mfcc for the three types of speech considered, a higher variance is noted that is associated with the clean speech mfcc relative to the low-noise and high-noise cases. In addition, Gaussian type behaviour is noted for all the coefficients, except for the first one where a bimodal distribution is apparent.

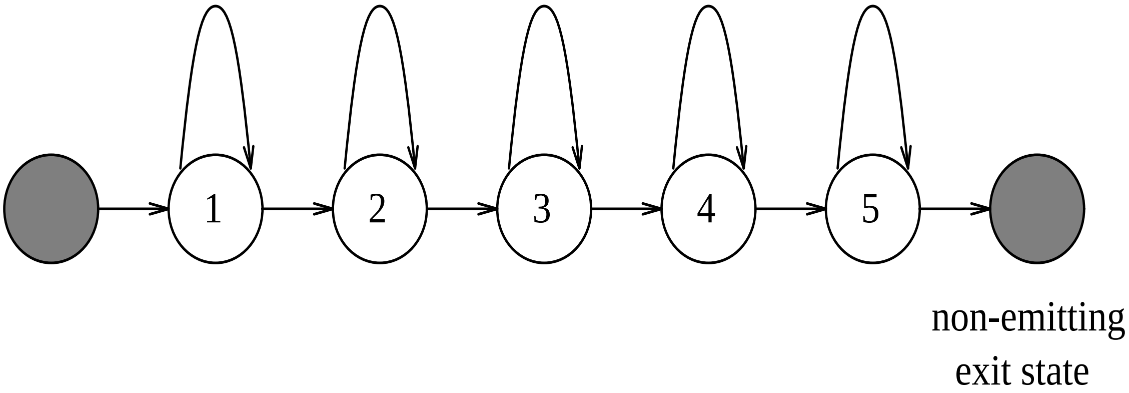


Figure 2: 5-state HMM with no skip transitions and non-emitting states at both ends.

A 5-state, discrete probability density HMM (figure 2) is used with an exit to a degenerate state. About 4500 10-th order mfcc training vectors are used to train the codebook by a clustering algorithm [33] into 64 prototype vectors. Mansour's beta distortion measure is used [34]. The baseline experiments described below using this distortion metric show its slight superiority over the Euclidean measure for real noisy speech. Each HMM is trained from 9 instances of a particular digit by the Forward-Backward algorithm [35]. Only 3 iterations are deemed sufficient for convergence. Recognition is done using the Viterbi algorithm [35], modified to take into account the exit to the non-emitting final state. The test set consists of 90 utterances, nine from each digit.

Because every experiment uses the same training and testing data sets, all the results are directly comparable. The experiments are carried out to see how various mismatch conditions affect the recognition results. Later, when different compensation schemes are applied to the different situations, the improvement in performance relative to those baseline results and the effectiveness of SBCDC in correcting that mismatch condition can be assessed. It can be argued that the mismatch situation high-noise train data/clean test case, to take an example, is an unrealistic task. However, the objective here is to apply SBCDC in such cases to find the performance limit of the transformation. In other related fields, an analogous situation would be a system trained on speech recorded by a low-quality microphone but then tested on a higher-quality microphone or in speaker adaptation a recognition system trained on 'difficult' speakers but then tested on a 'standard' one.

5.2 Baseline experimental results

The results (table 1) fall in line with other published results on this database [36, 37]. Best results are obtained when the train and test conditions match. The higher the SNR, the better the recognition results are. The terms in brackets show some of the results achieved for a Euclidean distortion measure in the VQ process with the other results barely affected, save for a 1% variation. Thus, the beta distortion metric will be used throughout whenever VQ is carried out.

	clean test (010-019)	low-noise test (046-055)	high-noise test (064-072)
clean train data (001-009)	**100.0**	64.5	20.0 *(16.7)*
low-noise train data (037-045)	80.0	**90.0**	18.9 *(15.6)*
high-noise train data (055-063)	43.3 *(32.3)*	35.6 *(33.4)*	**74.5** *(67.8)*

Table 1: Baseline results using mfcc as preprocessor and VQ beta distortion measure. The values in italics give the results using a VQ Euclidean distortion measure. The numbers in brackets identify the specific training and testing examples used throughout the experiments.

One of the recurring features of these preliminary recognition experiments seems to be that the digit *six* is never recognized for the low-noise case as training data. The most probable cause is segmentation errors during the hand labelling [2] and besides the digit *six* is notoriously difficult to segment in low SNR conditions. A look at the VQ distortion given by a representative set of vectors on a clean codebook indicates that the digit *six* suffers more than others in terms of average VQ distortion, especially for the low-noise data.

6 State-based cepstral domain compensation – *a priori* segmentation

This section starts by describing a simple digit-based compensation scheme, later extended to a more detailed state-based one. The details of the collection of state-based speech statistics and general points about the compensation scheme and its practical implementation are given. Empirical cepstral means shifts are derived from aligned HMMs of the corresponding clean and noisy speech. The potential benefits in carrying out SBCDC are enumerated. It should be emphasized that all the developments in this section are carried out under the implicit assumption that the segmentation points at which different sets of cepstral means shift operate are known *a priori*, and then used *a posteriori* in the handling of the mismatched speech vectors.

6.1 Simple whole-utterance cepstral means adjustment scheme

The combined Lombard and surrounding noise effects can be compensated, to some extent, by applying a simple cepstral compensation to the corresponding speech. For instance, average speech statistics for clean and noisy speech can be collected in a word-dependent fashion and used to disturb the clean training material to make it resemble noisy training material by applying a cepstral means shift on a vector-by-vector basis.

By using the whole utterance scheme described in section 3.3, the results in table 2 are obtained. A set of 10 adjusted HMMs are derived and are then aligned by the Viterbi algorithm to the test material. For the low-noise training material, we subtract low-noise means shift to obtain clean 'look-alike' speech. The same principle is applied to other train/test

[2] Recognition has been attempted using a one-state silence HMM at the beginning and end of the 5-state word model. There is hardly any improvement over the baseline [38]. Even after the silence models are removed and in the expectation that the speech material has been under-segmented, the utterances are segmented from a previous run of the Viterbi algorithm to get rid of the presumable begin and end silences (usually found to be no more than 3 frames) but the results are not improved.

conditions. The nominal SNR is assumed known for the test material. Calculation of Percent Recovery or error rate reduction (e.r.r) is given from the error rate (E.R) by:

$$\text{Percent Recovery} = 100\% \frac{E.R_{\text{Baseline}} - E.R_{\text{Compensated}}}{E.R_{\text{Baseline}} - E.R_{\text{Matched}}}. \tag{43}$$

Compared to the baseline performance, table 1, there is a measurable improvement in performance with better results for the high SNR test material than with the low SNR test material.

	clean test	low-noise test	high-noise test
clean train data	**100.0**	71.1	31.1
low-noise train data	87.8	**90.0**	33.3
high-noise train data	68.9	51.1	**74.5**
clean train data		26	20
low-noise train data	39		26
high-noise train data	45	28	

Table 2: Recognition results in the upper half using digit-dependent whole-utterance cepstral means adjustment of training material. The lower half indicates the error reduction rate relative to the baseline matched training and testing conditions.

6.2 Collection of state-based cepstral means shift

The word-dependent, state-dependent statistics are collected as follows: Word HMMs are trained on the clean and noisy speech material separately. The corresponding observation sequences are then time-aligned with the corresponding word HMM using the Viterbi algorithm. Speech vectors associated with each state of the HMM are collected. The relatively low variances associated with the average normalized state duration suggest that the speech material, whether clean or noisy, divides itself well into stable regions of each state of the HMM. Averaging of the mfcc vectors in each state and subtraction of the clean from the noisy average yields a digit-dependent and state-dependent cepstral means shift vector. Whenever cepstral means adjustment is applied to the speech material, the average normalized duration of a particular state multiplied by the total number of frames in the observation sequence for that particular word yields the number of frames that a particular set of cepstral means is applied. Some speaking rate normalization is achieved in the same step. The whole process is summarized in figure 3.

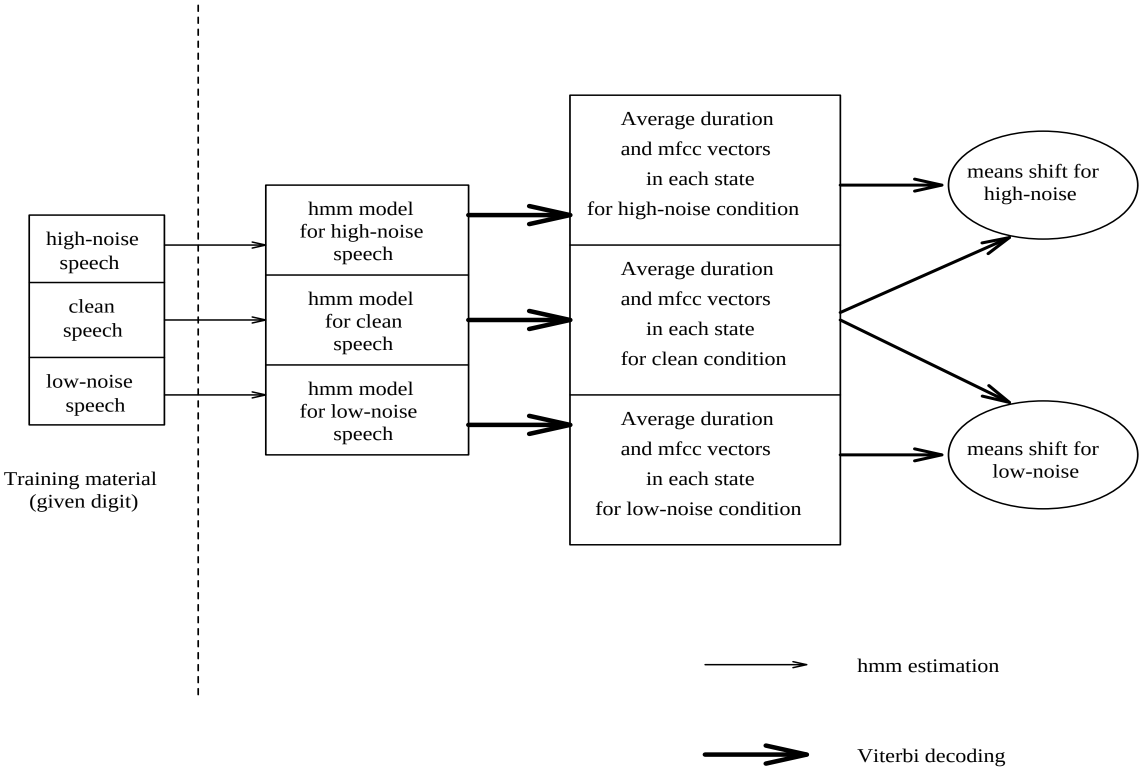

high-noise speech
clean speech
low-noise speech
Training material (given digit)
hmm model for high-noise speech
hmm model for clean speech
hmm model for low-noise speech
Average duration and mfcc vectors in each state for high-noise condition
Average duration and mfcc vectors in each state for clean condition
Average duration and mfcc vectors in each state for low-noise condition
means shift for high-noise
means shift for low-noise
hmm estimation
Viterbi decoding

Figure 3: Implicit state-to-state alignment for the estimation of average mfcc vectors and state durations.

Table 3, 4 and 5 show for some digits the actual average normalized duration in each state for each of the three types of speech.

digit	dur.	var.
zero	0.26,0.18,0.22,0.20,0.15	0.012,0.006,0.005,0.010,0.002
one	0.28,0.18,0.16,0.33,0.05	0.006,0.001,0.000,0.005,0.001
two	0.26,0.13,0.30,0.19,0.12	0.002,0.002,0.002,0.002,0.006
..	...	..
nine	0.23,0.12,0.19,0.16,0.14	0.012,0.002,0.011,0.003,0.008

Table 3: Average normalized state durations and variances for clean speech for a 5-state HMM.

digit	dur.	var.
zero	0.26,0.16,0.03,0.40,0.14	0.010,0.001,0.000,0.025,0.012
one	0.28,0.15,0.14,0.27,0.16	0.014,0.002,0.003,0.011,0.010
two	0.28,0.10,0.24,0.15,0.23	0.003,0.007,0.003,0.005,0.008
..	...	..
nine	0.24,0.08,0.13,0.10,0.24	0.010,0.000,0.003,0.003,0.008

Table 4: Average normalized state durations and variances for low-noise speech for a 5-state HMM.

digit	dur.	var.
zero	0.24,0.15,0.19,0.29,0.12	0.008,0.013,0.015,0.016,0.004
one	0.22,0.24,0.18,0.10,0.27	0.009,0.012,0.020,0.008,0.043
two	0.29,0.14,0.12,0.32,0.13	0.014,0.002,0.006,0.015,0.007
..	...	..
nine	0.21,0.30,0.11,0.18,0.22	0.006,0.004,0.001,0.008,0.032

Table 5: Average normalized state durations and variances for high-noise speech for a 5-state HMM.

Viterbi decoding applying N-state HMMs to an observation sequence allows the identity of particular frames that are attached to each of the N states to be known. Each of these N states will presumably represent statistical regions of speech that will be different from the adjacent states. The objective of state-based cepstral (means) domain compensation (SBCDC) is to change each mfcc vector attached to a particular state, or hypothesized to be attached to a particular state, to that value that might have been observed if the underlying speech event had occurred in different ambient conditions and noise levels. On the other hand, the generalization to other SNRs may at first seem restricted and those cepstral means may be too data-dependent and speaker-dependent. These concerns will be addressed in the coming sections except the speaker-dependency of the cepstral means.

This state-to-state alignment has been implemented in an identical way in Shinoda's work [39]. Standard speaker's HMMs are used in Viterbi algorithm to segment training data for a new speaker. The feature vector for each frame period is time-aligned with an HMM state, and the mean vector for each HMM state is replaced with the average of those feature vectors which have been time-aligned to the state.

training material	recognition results on training material
clean:(001-009)	100.0
low-noise:(037-045)	100.0
high-noise:(055-063)	100.0

Table 6: Recognition tests on training material.

The recognition results when doing the state-to-state alignment on the training material are shown in table 6. Because the results are perfect, high confidence can be placed in the Viterbi decoded state sequence. This method of state-to-state alignment has been made possible because of the relative low confusability of the digit vocabulary – had there been a wrongly decoded model (i.e. an incorrect recognition), the state-to-state alignment for that particular observation sequence would have to have been modified by either forgetting that sequence or else going to its correct model but incorrectly Viterbi decoded and then collecting the state information. Three different types of cepstral means deviations, acting as a measure of the noise in the mel-scale are derived, namely clean-high means shift, clean-low means shift and low-high means shift.

An informal visual observation of the digit-dependent, state-dependent means shift has revealed some interesting points:

- More detailed variation of means shift for each state, as compared to only digit-dependent case. It is precisely the detailed nature of these means shift that will allow a more accurate compensation to a particular condition of speech.
- The same pattern of increase and decrease is observed along the mfcc index number for both high-noise speech and low-noise speech relative to the clean speech, although the absolute changes of these means shift tend to vary. This suggests a simple scheme to attack noisy speech recorded in a car between 0 and 100 Km/h by interpolating the cepstral means shift values obtained. A later experiment confirms the plausibility of this approach.

To summarize, the potential benefits of SBCDC will be:

1. Speaking rate normalization. Lombard effect is implicitly accounted for.
2. No localized frame by frame SNR calculation – only the nominal SNR of incoming signal needed.
3. Means shift is a simple vector subtraction (addition).
4. Averaging of vector statistics and state duration to calculate HMM segmentation points smooths out 'outlier' vectors in a particular utterance.
5. Correlation among frames connected to a particular HMM state is approximately preserved. The average timing information is kept in the compensation process.

If training data is present, further benefits may apply like:

1. No need to know subword boundaries explicitly. Suitably emphasized acoustic events are systematically captured by Viterbi alignment.
2. Collection of speech vectors is carried out in such a way that is effective for recognition on the training data.
3. Compression of information is very significant. Each word is characterized by N (5 used) sample averaged cepstral vectors and N average normalized state segmentation points. If 9 utteranes from a particular digit is used, this represents an average reduction factor of about 90 from the frame vectors to this format.
4. Averaging statistics over several utterances smooths out some 'outlier' utterances over the whole data set.

6.3 Cepstral means adjustment on train data

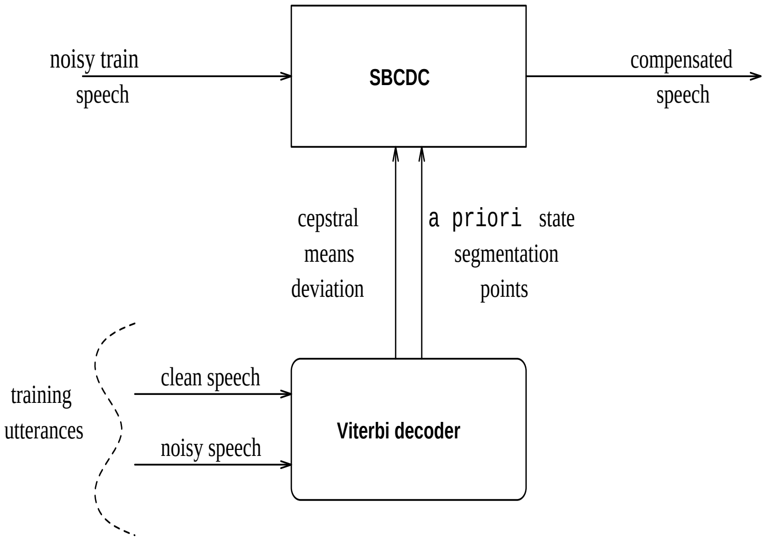


Figure 4: SBCDC on training material (clean train/high-noise test case).

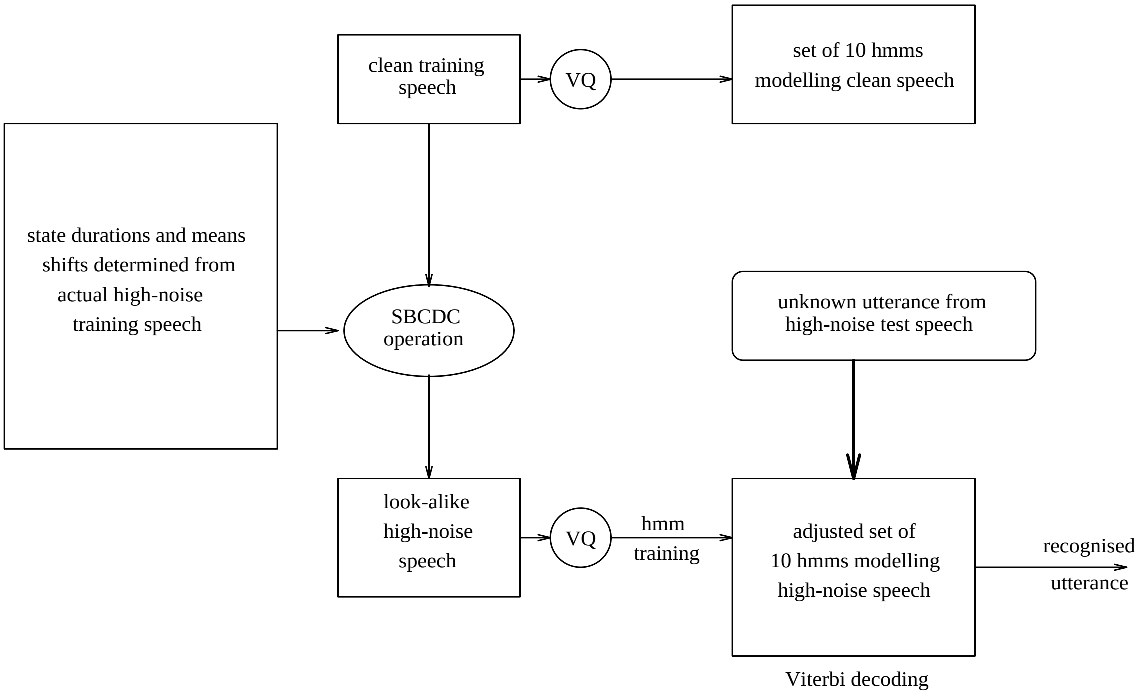

clean training speech
VQ
set of 10 hmms modelling clean speech
state durations and means shifts determined from actual high-noise training speech
SBCDC operation
unknown utterance from high-noise test speech
look-alike high-noise speech
VQ
hmm training
adjusted set of 10 hmms modelling high-noise speech
recognised utterance
Viterbi decoding

Figure 5: Detailed operations involved for SBCDC on training material (clean train/high-noise test case).

A schematic overview is given in figure 4 and the detailed operations in figure 5. Table 7 is obtained when the method described is directed to the training data. To review the techniques once more, consider the high-noise training data from which is subtracted the high-noise means shift to obtain training material that 'resembles' clean speech (third row, first column of results in table 7). Also, from the high-noise train data is subtracted the high-low means shift to obtain a low-noise speech prototype (third row, second column in table 7). The recognition improvement is highly significant now, especially the low-noise train/clean test and high-noise train/clean test cases. The recognition improvement is not as great for the clean train/high-noise test case.

	clean test	low-noise test	high-noise test
clean train data	**100.0**	80.0	51.1
low-noise train data	98.9	**90.0**	60.0
high-noise train data	100.0	74.5	**74.5**
clean train data		61	57
low-noise train data	95		74
high-noise train data	100	72	

Table 7: Digit-dependent state-based cepstral means adjustment of training material in an attempt to match the speech condition of the test material; testing material unchanged. The lower half indicates the error reduction rate relative to the baseline matched training and testing conditions.

The low-noise train/clean test and high-noise train/clean test (remember that means shift vectors are subtracted from the training material and VQ re-applied to the compensated training material) are highly remarkable results. Firstly, the SNR has been considered nominal even though the actual frame-by-frame localized SNR can be highly varying. Secondly, the non-negligible variances associated with the HMM segmentation points imply that even after cepstral compensation is incorrectly applied to a particular region of the utterance concerned for a few frames, it does not affect the modified statistics of the compensated material and its modelling by an HMM when time-aligned with the test material. This suggests a certain robustness of SBCDC provided the right set of digit-dependent cepstral means shifts is applied.

The e.r.r for clean train/low-noise test, clean train/high-noise test and low-noise train/high-noise test are fairly respectable but this time the noise vectors are added to the training material and VQ re-applied to the compensated material. When noise in the form of cepstral means shift vectors are added to the clean or low-noise speech, an attempt is made to mask some of the low-amplitude speech events. Significant probabilities are then assigned to all VQ labels corresponding to the silence-type acoustic space. This will hopefully include those labels that are not observed during the training session. Overall, the results are very satisfying because the more detailed state-based scheme gives a higher normalised MSE reduction (figure 6) than the whole-utterance compensation of the last subsection 6.1.

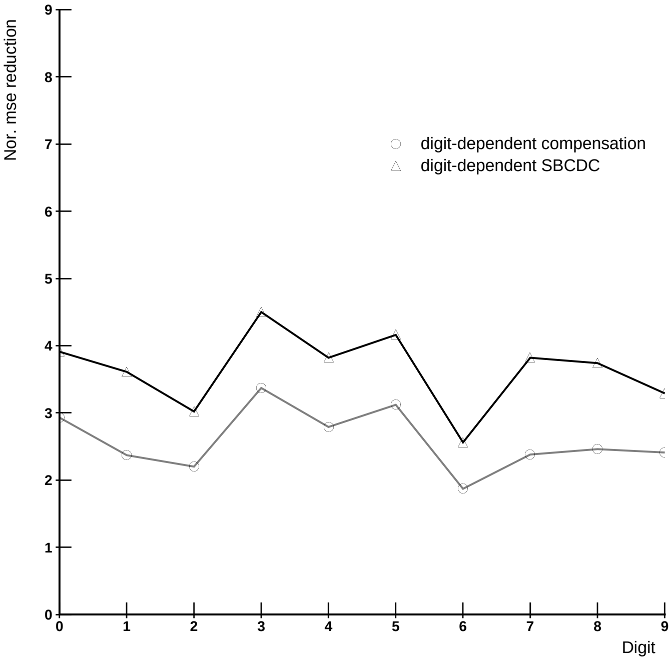


Figure 6: Normalised MSE reduction for whole-word compensation and state-based compensation for the different digits in the application of cepstral means shift on the training material. The high-noise train/clean test case is considered here.

Lack of training data usually entails too much sharpness in the estimates. This is always the inevitable fact in speech recognition that the statistical properties of the training data cannot be guaranteed to be that of the test data. This is made worse at low SNR. One important point to verify is to compare the recognition rate when the HMM is used on both training and independent test data. The difference in recognition accuracy obtained is large – 100% recognition for the training set and 74.5% for test set in the case of clean train/high-noise test baseline case. It implies that there is not enough data in the training phase. This should be contrasted with 100% recognition from the training set and 100% on the testing set for the matched clean train/clean test case. This point is further belaboured by the fact that reversing the role of the training and testing set the same imbalance in the recognition results is achieved (100% from the training set formerly the test set, and 73.3% from the test set formerly the training set). This small sub-experiment preempts the claim that consistent ‘bad’ data can be said to exist in the current high-noise test set.

7 Cepstral means adjustment on test data – *a priori* segmentation

Application of cepstral means shift to the testing material is more problematic. Because the test speech is by definition unknown, which digit-based means shift to apply is unknown, whereas for the training material the identity of a given utterance is known. A hypothesis-driven method is needed: To an unknown test utterance, all the word-dependent means shifts are applied giving rise to V hypothesised but acoustically compensated utterances (where V is the number of words in the vocabulary). A Viterbi alignment is carried out against the reference model of the hypothesis. The overall maximum likelihood result is selected. Because the identity of the applied word means shift is known, the unknown utterance identity is inferred. A schematic overview is given in figure 7 and the detailed operations are given in figure 8.

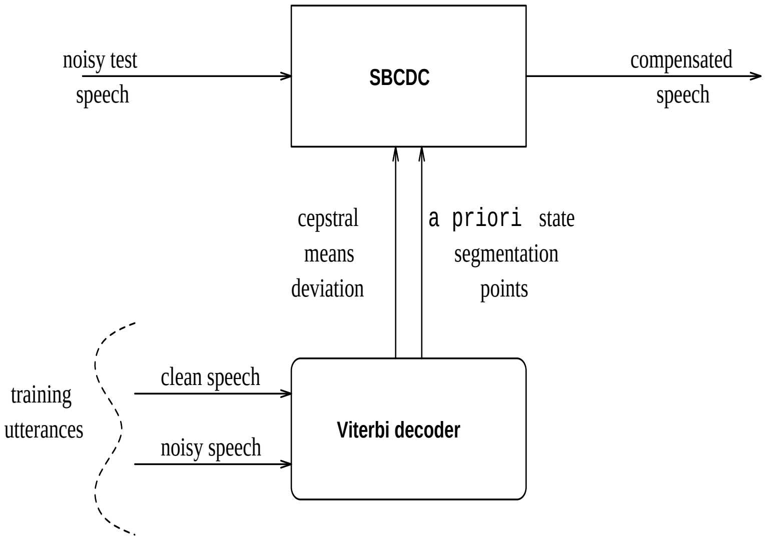


Figure 7: SBCDC on test material.

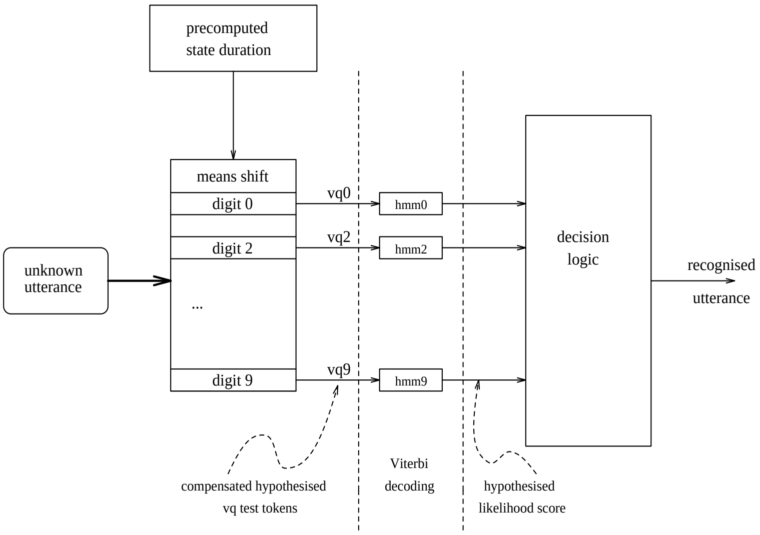


Figure 8: Detailed operations of SBCDC on test material.

Although the hypothesised utterance might deviate from the desired cepstral means compensated utterance, as long as the separation between this hypothesised utterance and the remaining ones is large in the cepstral domain, it will be possible for it to be better time-aligned and better matched against the hypothetical reference HMM than the remaining hypotheses and thus provides the highest likelihood score. The crux of the matter lies in the fact that the identity of the given speech is unknown and the best conjecture is that a given transformation will drive the correctly matched noisy cepstral vector (argument for clean train/high-noise test case) to a cepstral space that is highly different from the transformed incorrectly matched noisy cepstral vectors and closer to the cepstral space of the correctly matched digit. Nadas argued similarly for his adaptive labeling algorithm [22].

7.1 Experimental results

	clean test	low-noise test	high-noise test
clean train data	**100.0**	84.5	40.0
low-noise train data	92.2	**90.0**	56.7
high-noise train data	56.7	80.0	**74.5**
clean train data		56	25
low-noise train data	122		53
high-noise train data	43	114	

Table 8: Training material unchanged; digit-dependent state-based cepstral means adjustment of testing material in an attempt to match the speech condition of the training material. The lower half indicates the error reduction rate relative to the baseline matched training and testing material.

An artificial situation is assumed whereby the *a priori* SNR is known i.e. whether the test material comes from clean, low-noise or high-noise speech is known.

The improvement brought about by SBCDC on the training data cannot be directly compared to SBCDC on the test material. Consider the high-noise train/clean test case. SBCDC on the training data yields the compensated clean train/clean test situation. This result is contrasted with the matched clean train/clean test case. On the other hand, SBCDC on the test data will yield high-noise train/compensated high-noise test case. This result is contrasted to the matched high-noise train/high-noise test case.

Remember how SBCDC on test material is carried out – a hypothesis is made about which digit-dependent shift is applied and the compensated speech is aligned with the HMM of the hypothesis. The clean train/low-noise test and low-noise train/high-noise test case performance holds remarkably well. There is a complete dip in performance for the clean train/high-noise test case, and this is disappointing because it corresponds closer to real-life application. SBCDC using state points and means shift derived from the training material works less well when directed on the high-noise test set. The lower performance of the clean train/high-noise test case relative to low-noise train/high-noise test is partly explained by the fact that the low-noise train material contains some information about the engine noise whereas the clean train material contains no such knowledge in either the VQ codewords and in the HMM B probabilities modelling the training material.

As mentioned in the discussion part of section 3, the reference HMM B matrix can provide the average mfcc vector needed in each state by using (36) in subsection 3.2. 37.7% has been obtained for clean train/high-noise test case. This is marginally lower than the 40% obtained for the same set of data when SBCDC is carried out in the usual way.

7.2 Beyond simple means shifting

In doing the means shift operation, the transformation is attempting to maximize the similarity between the noisy and clean speech. Second-order statistics in the form of variance information can also be used. A more general transformation where some form of variance equalization is performed before means shift can be applied to a speech vector y is represented by:

$$T(y) = \mathbf{A}y + \mathbf{b}. \tag{44}$$

Observe that the above equation reduces to simple means shifting with $\mathbf{A}$ being the identity matrix. Roucos showed how to estimate $\mathbf{A}$ and $\mathbf{b}$ from the covariance matrices of the two different types of speech [24].

A preliminary experiment whereby a simple variance ratio equalization (a naive state-based simplification of (44) with non-unity diagonal elements) followed by the means shift for the clean train/high-noise test case of SBCDC on the test material leaves the recognition score (40% from table 8) unchanged. Although the above transformation is more sophisticated, there are several reasons which might make it inappropriate for use. For example, densities are not really Gaussian and increasing the overlap in cepstral space may not necessarily yield more similar VQ segments. But above all, proper calculation of the covariance matrix necessitates a fair amount of data and the matrix probably has to be pooled across all the digits thus losing the specificity of the simple digit-dependent means shift transformation.

7.3 Use of more states

Perhaps finer cepstral means adjustment by using more states can lead to higher recognition accuracy for SBCDC on the testing material. Finer segmentation into more states may augment the better-tracking and detailed characteristics of the noise process. On the other hand, influence of inaccurately estimated state segmentation boundaries will increase. Indeed, over-segmentation of some unvoiced sounds such as phoneme /s/ for digit *six* will lead to a large segment length variance and outlier speech vectors will be more common affecting adversely the averaging of the associated speech vectors. This is verified by the experimental evidence for clean train/high-noise test in which state-based compensation is applied to the test material resulting in:

No. of states	Recognition accuracy (%)
3	38.9
5 (table 8)	40.0
7	37.7
10	34.4

Table 9: Recognition accuracy with increasing number of HMM states (clean train/high test case).

A priori cepstral means vector collection has to be carried out again for the other cases. When doing the collection of data, the average variance of state occupancy is, as predicted, larger per number of states.

7.4 Correct hypothesization

To obtain a performance limit to this type of adjustment on the test material, the following additional experiments were carried out (refer to figure 9). Experiments on the matched training and testing conditions were not pursued. The objective is to assess what difference a correct hypothesis makes to SBCDC on the test material, and ultimately to find the upper limit of SBCDC on the test material.

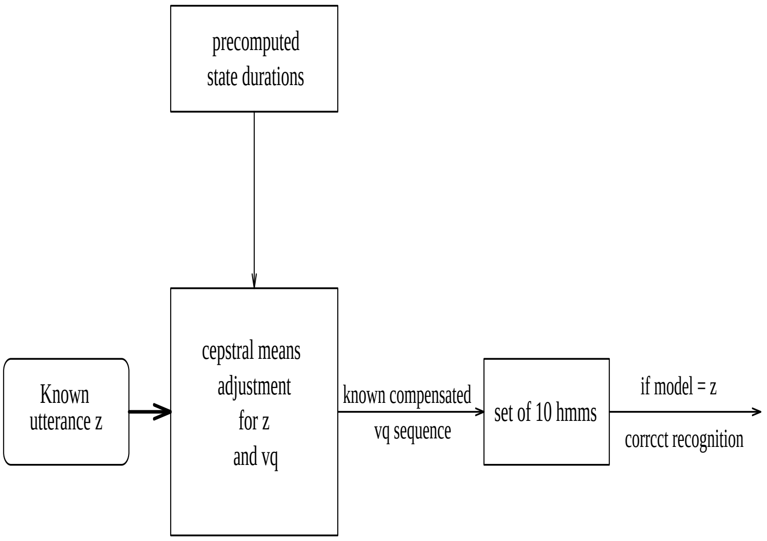


Figure 9: SBCDC on test material (ideal case) and correct hypothesization scoring. The set of 10 HMMs are from the reference training material.

	clean test	low-noise test	high-noise test
clean train data		90.0	100.0
low-noise train data	92.2		95.6
high-noise train data	63.4	81.1	
clean train data		72	100
low-noise train data	122		108
high-noise train data	64	117	

Table 10: Training material unchanged; digit-dependent SBCDC on test material (ideal case) in an attempt to match the speech condition of the training material. The lower half indicates the error reduction rate relative to the baseline.

The results are shown in table 10. The recognition results are obviously better than in table 8. Consider the clean train/low-noise test case whereby the low-noise test digit is correctly compensated, but this time the e.r.r is 72% compared to 56% for SBCDC on test material. To some extent, cepstral compensation is not adequate and some correctly compensated digits align themselves better against incorrect reference models. A similar conclusion holds for the high-noise train/clean test case. This result is reversed for the low-noise train/high-noise test case where the correctly compensated high-noise test digit, when matched against the reference HMMs, perform better than the matched low-noise train/low-noise test condition (108% e.r.r). Some pre-empting of the test conditions has been achieved by SBCDC on the high-noise test material.

The perfect score obtained for the clean train/high-noise test case indicates that whenever state-based means shift is done on a particular digit, it is well recognized but that, in reality, other digit-based transformations may yield higher likelihood ratios. A known high-noise test digit is mapped by SBCDC into an absolutely certain (identity-wise) compensated clean digit. The latter always aligns to the correct reference HMM, but table 8 indicates that an incorrectly matched, hypothesized compensated speech sequence, aligned with its incorrectly matched reference HMM may yield a higher likelihood. The compensation, as currently implemented, even with correct hypothesization is simply not accurate enough.

7.5 Cepstral means adjustment simultaneously on both training and testing material – *a priori* segmentation

Some channels in a filterbank speech spectrum that are of low amplitude levels will be more affected by noise than other channels. Klatt proposed substituting the channel outputs below a certain threshold by a mask noise level in both the reference and test speech frames [40]. This will attenuate the acoustic distortion accumulation. Moreover, the addition of noise masks out most of the silence-like speech frames which are dependent on the ambient noise level and also may mask out low-level events present in a clean environment but absent in a noisier one [41].

All the previous experiments are concerned with bringing the statistics of either training material to its matched testing material and vice versa. Bringing the statistics of both the training material and testing material to a nominally common SNR level represents a way to ensure robustness because the commonness of the statistical transformations on the two sets of speech material can bring some regions of the utterance to identical cepstral shapes. Furthermore, for the test material some VQ regions not observed during training are mapped to those common cepstral shapes. Because of the average values associated with the segmentation points and means shift in SBCDC, the common mapping will be from a varying distribution statistics to a more compact one. Whereas Klatt [40] and Compernolle [42] added 'noise' to both test and training material, one work where both speech train and test data sets are brought iteratively by autoregressive filtering to a higher SNR level is Ephraim's earlier work [43]. The following results are shown in table 11.

	clean test	low-noise test	high-noise test
clean train data	**100.0**	84.5	40.0
low-noise train data	98.9	**100.0**	45.6
high-noise train data	100.0	84.4	**51.1**
clean train data	**0.0**	56	25
low-noise train data	95	**+10.0**	33
high-noise train data	100	76	**-23.4**

Table 11: Digit-dependent SBCDC simultaneously on both train and test material to clean condition. The lower half indicates the error reduction rate relative to the baseline score of matched clean train/clean test condition. Bold term in lower half indicates absolute level of recognition relative to its proper matched condition.

The low-noise train/clean test and the high-noise train/clean test are left untouched because that represents the objective and results are simply quoted from table 7. Similarly, the clean train/low-noise test and clean train/high-noise test have been carried out (results in table 8) as there is no further need to apply SBCDC to the clean training material.

When the low-noise train/low-noise test case is mapped to compensated clean train/compensated clean test, the longstanding misrecognition of digit *six* disappears – the recognition results are even better than the matched low-noise train/low-noise test conditions. Thus, for low-noise speech material (medium SNR), SBCDC on both material gives better results by mapping them to some common compensated higher SNR level statistics. However, the mapping from low-noise train/high-noise test to compensated clean train/(hypothesised) compensated clean test is now low (33% e.r.r) compared to the baseline matched clean train/clean test condition. This result is amplified with compensation on the matched high-noise train/high-noise test where a reduction of 23.4% absolute recognition accuracy is observed.

Many factors may account for this poor behaviour at such low SNRs: it is already apparent from the previous sections that SBCDC on the test material does not provide substantial correction. What happens for the low-noise test (second column) is that maybe at a medium SNR, a mapping of the speech material to a higher SNR would be advantageous: more data at mild conditions ought to be analyzed before a definitive conclusion can be raised. A quirkiness of the implementation of SBCDC to give 100% for low-noise train/low-noise test may not be discounted; compensation applied to digit *six* that has been erroneously segmented might have corrected some boundary effects and perhaps no significant breakthrough can be expected in this direction.

8 Robustness of the cepstral means shift technique

This section will investigate the applicability of the cepstral means shift to other SNRs. Because noisy speech at arbitrary SNRs is not available, the basic set of noisy data compensation vectors at 0 Km/h is used to predict the compensation at 100 Km/h and recognition results assessed to check the validity of the approach. Similarly, the state segmentation points along the utterance are SNR-dependent and have been calculated by averaging over the entire training set. The fact that cepstral means adjustment to the training material works remarkably well especially for the non-matching noisy train/clean test conditions, even if those segmentation points are average quantities with non-zero variances confirms the robustness of those pre-calculated segmentation points. Indeed, fairly good results even with uniform segmentation points, as demonstrated later, is further evidence to support this. The experiments will be restricted to applying the compensation to the test material as high recognition accuracy has been achieved through the compensation scheme directed to the training material. The clean train/high-noise test case will be the particular focus as it represents a more realistic scenario.

8.1 Fitting regression lines to cepstral means shift

An attempt is made to check whether the cepstral means adjustment is applicable at various SNRs (to some extent speed-dependent) by having a basic set of values and adjusting those values by a factor proportional to the SNRs – because the variations of cepstral means deviation relative to the clean speech, either for the whole digit or state-based, follow roughly the same pattern along the mfcc coefficient index.

It will be very convenient if to the cepstral means shift at 0 Km/h can be added some SNR-dependent but constant value for coincidence with the one at 100 Km/h. However, in practice, this does not apply and the next best attempt for best fit after addition of some SNR-dependent value is straight line regression analysis. From the cepstral means shift at 0 Km/h on the x-axis and the ones at 100 Km/h on the y-axis, the regression lines (gradient $\hat{m}_{s_i}$ and intercept $\hat{c}_{s_i}$ for state i) for each of the five states are estimated:

$$\mathbf{y} = \hat{m}_{s_i}\mathbf{x} + \hat{c}_{s_i}.$$

Averaging all the gradients and intercepts from each state gives the average gradient and intercept denoted by $\hat{m}_{s_{av}}$ and $\hat{c}_{s_{av}}$ respectively. The final digit-dependent transformation applied to a cepstral means shift at 0 Km/h to yield the predicted cepstral means shift at 100 Km/h is given by:

$$\mathbf{y} = \hat{m}_{s_{av}}\mathbf{x} + \hat{c}_{s_{av}}.$$

The averaging is done to make the prediction of cepstral means shifts more general and the resulting values are applied to the cepstral means shift of the test material to check how well those values perform. The results for the low-noise train/high-noise test case are obtained using the low-noise train material as the base (table 12):

Training/testing conditions	Recognition accuracy (%) using predicted shifts	Recognition accuracy (%) using empirical shifts
clean train/high-noise test	35.6	40
low-noise train/high-noise test	53.3	56.7

Table 12: Recognition accuracy using digit-dependent predicted cepstral means shift on the test material (first column) compared to the empirically determined cepstral means shift (second column).

The recognition results are marginally lower relative to the ones with empirical means shift but are encouraging. This subsection as well can easily be evolved into a simple method for indicating approximately the overall energy shifts in each of the states if SBCDC for log energy shifts is monitored.

8.2 Robustness of the segmentation boundaries

In general, the pre-determined state segmentation points vary over quite a wide range at the different SNRs. In addition, some of the normalized state durations have non-negligible variances associated with them. But some small subset experiments suggest that even with uniform linear segmentation i.e. into five equal durations, the recognition results are not affected that adversely. A similar conclusion is obtained using segmentation boundaries from the clean training set and applied on the noisy test material. This implies a robustness of the segmentation points that could be used over different SNRs and crudely implies that the duration of the region covered by a state of the HMM is, in the main, independent of the SNR condition, at least as far as the particular speaker-dependent digit experiments show.

Clean train/high-noise test conditions	Recognition accuracy (%)
Baseline	20.0
SBCDC on high-noise test (table 8)	40.0
Linear segmentation	32.2
Segmentation from clean train set	36.7

Table 13: Recognition accuracy using differing segmentation boundaries (clean train/high-noise test case).

Similar conclusions are reached for the low-noise train/high-noise test case, as attested by the following results in table 14.

Clean train/high-noise test conditions	Recognition accuracy (%)
Baseline	18.9
SBCDC on high-noise test (table 8)	56.7
Linear segmentation	47.8
Segmentation from low-noise train set	52.2

Table 14: Recognition accuracy using differing segmentation boundaries (low-noise train/high-noise test case).

The results imply that Lombard effects associated with low SNR speech are not that significant when carrying out the Viterbi decoding equations although the differences in average state durations in clean and high-noise speech varies markedly. More data are required for the generalization of this finding.

9 Cepstral means compensation – unknown segmentation

Application of means compensation technique to test data has, so far, required a hypothesis-driven approach, known *a priori* segmentation points and speech samples characterizing the test signal. This section presents an algorithm whereby the state segmentation points and compensation vectors are determined from a single test sequence. The method is iterative but still requires a hypothesis-driven treatment of the data. It corresponds to the full implementation of the E-step and the M-step described in subsection 3.1.

9.1 Iterative cepstral means shift algorithm

For the testing utterances, a copy of both the mfcc vectors and the VQ counterparts are required. For a schematic overview, refer to figure 10. The detailed operations are shown in figure 11.

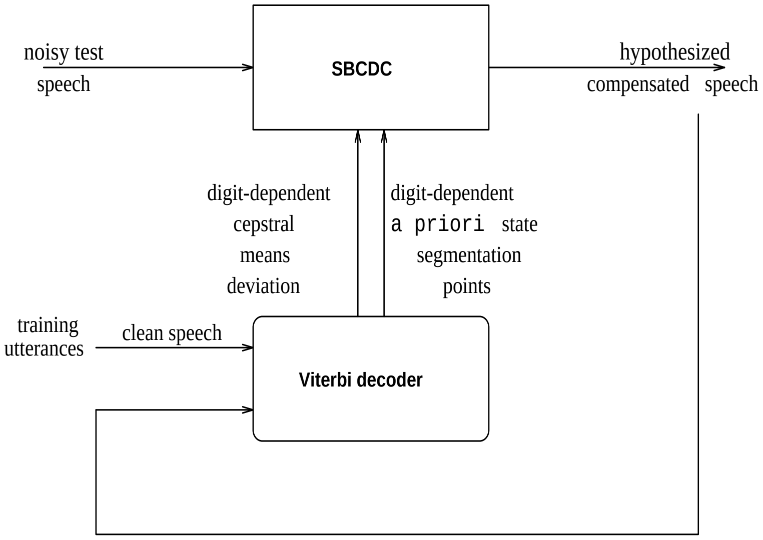


Figure 10: Iterative SBCDC on test material.

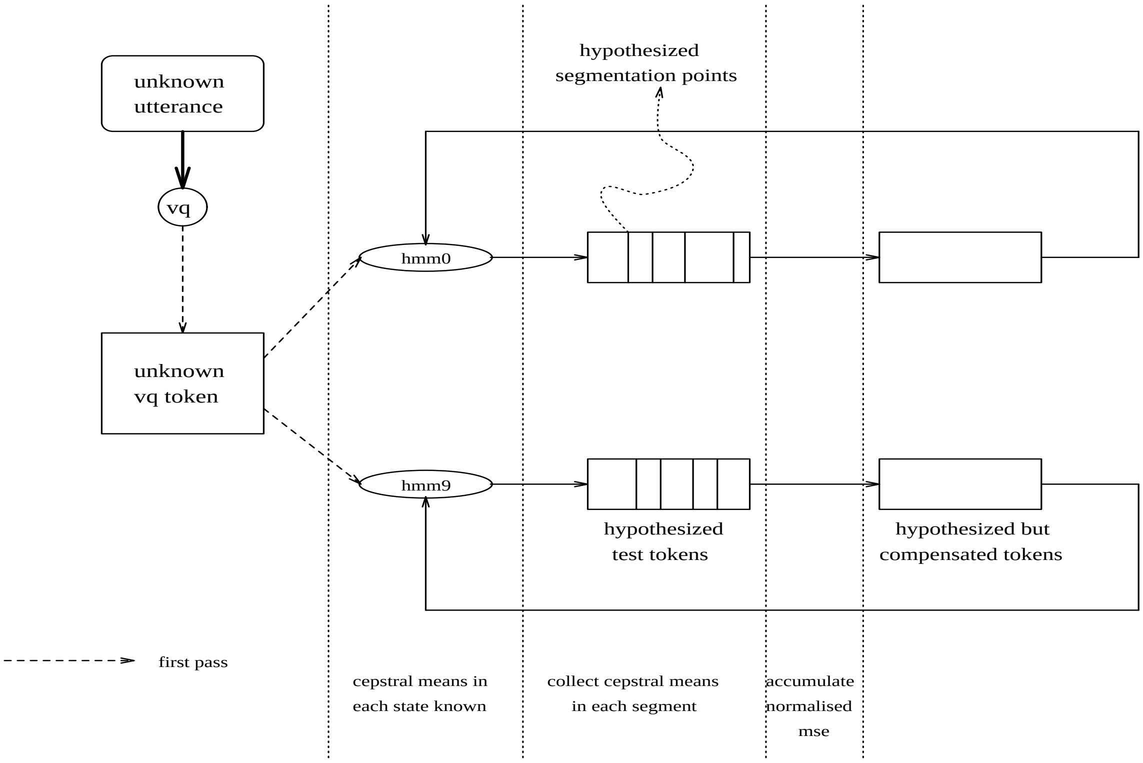


Figure 11: Detailed operations involved in iterative SBCDC on test material.

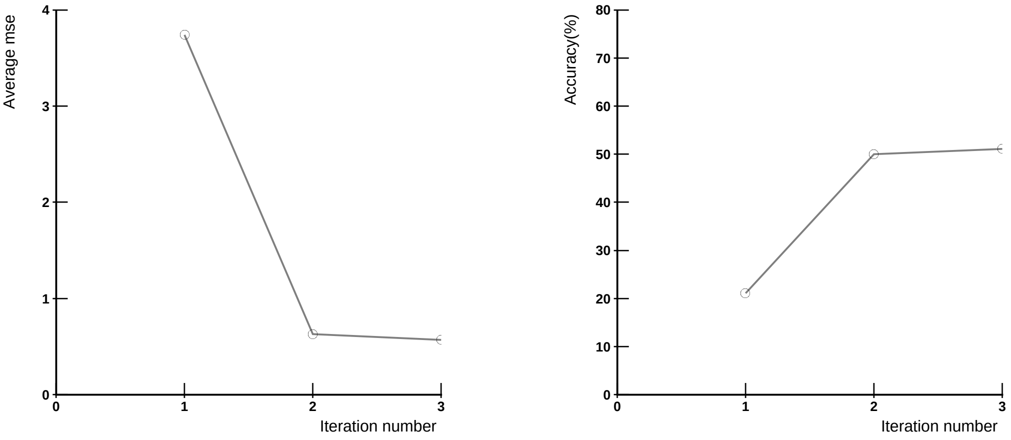
The iterative SBCDC algorithm involves the following operations:

- VQ encode the unknown test token using the codebook provided by the training data.
- For each of the reference HMM digits, a Viterbi alignment is done against the unknown VQ token. By using a particular reference HMM, a hypothesis is made that the unknown

VQ token has the identity of that reference HMM. For each of the hypothesised reference HMM, calculate the state segmentation points for that token to stay in the different HMM states. Keep track of the log likelihood in the Viterbi alignment. If the threshold for collected MSE is attained, stop iteration and infer identity of unknown token by maximum likelihood criterion.

- Otherwise, collect the cepstral means from each state (obtained from the cepstral vectors from the utterance and the specific region covered by that state). Calculate the cepstral means for each state in the hypothesised HMM. Calculate the cepstral means disturbance vector and from the specific region covered by that state, subtract the disturbance at the vector level. Repeat for all the ten hypotheses. Keep track of the length-normalized MSE.

In summary, the iterative SBCDC algorithm can be interpreted as a removal of the existing mismatch caused by car noise between the input signal and a N-vector format word model representing the (clean) baseline speech signal by state-based cepstral subtraction. Because the conditions of the unknown VQ token from low or high noise case are so different from the clean reference HMMs, there will be gross segmentation errors in the splitting of the state track in the first pass. If the identity of the input VQ token is known, the above iterative procedure can easily be shown to converge locally and the problem crops up in many speaker normalization techniques.



Average mse for iterative SBCDC

Recognition accuracy for iterative SBCDC

Figure 12: Recognition accuracy and average MSE with iteration number for iterative SBCDC on test material: low-noise train/high-noise test case. Overall recognition accuracy is 51.1%.

Figure 12 is concerned with the monitoring of MSE with the number of passes for a correctly recognised digit zero. One pass is sufficient in practice as the overall recognition accuracy hardly changes afterwards. Even when an input token is aligned against an incorrectly matched reference model, the decrease in MSE occurs, although not sharply. There is an improvement of $(51.1 - 18.9)/(90.0 - 18.9) = 45\%$ in error reduction rate for the low train/high-noise test case. But still a slightly higher error rate reduction has been achieved with SBCDC on test data with *a priori* state segmentation points (53%).

The results obtained with the recognizer trained and tested in similar environments provide an approximate upper bound on the potential of the noisy speech recognition task. It is possible to envisage a situation whereby noisy speech recognition operating in a dynamic mode could exceed the matched noisy speech recognition results simply by being able to gather more data. However, for all the reported experiments, the amount of data available for the noisy speech recognition in iterative SBCDC (1 utterance) is never more than that available for non-iterative SBCDC (9 utterances), therefore the upper bound mentioned is absolute in this case.

In Furui's work [28], effectiveness of adaptation has been mainly evaluated using spectral distortion. He found that the spectral distortion and recognition error rate are highly correlated. A comparable observation is obtained here. The lower recognition results with only one utterance to determine the state segmentation points and cepstral means shifts represent a finding which is not surprising. The statistical distribution of the corrected speech (hypothesized) cannot be guaranteed to be close to that of the training set and hence that of the reference HMMs. In Acero's work [17], the equalization and noise factor estimates required for the correction of mismatched speech vectors are found to exhibit a large variance for short utterances. Similarly, in Ephraim's speech enhancement scheme [12], good quality of the enhanced speech is obtained when decoding is done using the clean speech to obtain the state sequence and sometimes bad quality of the enhanced speech is obtained when decoding is performed using only low SNR speech as bootstrap.

10 Discussions

Because of the limited data for both training and testing, the low confusability of a clean digit vocabulary set and the speaker-dependent restriction, only broad generalizations can be offered. One thing must be emphasized at the outset: the speech database that has been used is a particularly difficult one. The high-noise speech is estimated at a nominal -7 dB level whereas if the study in the ARS report [36] is referred to for baseline recognition performances on other similar databases, higher levels of performance have been obtained. Possible extensions of the work are also mentioned.

An EM framework is used on the complete data consisting of the set of baseline signal, mismatched signal and the state sequence through an N-state Hidden Markov source model. The incomplete data consists of the observed mismatched signal. The parameters to be estimated are the set of state-based compensation vectors. The relationship between the complete and incomplete data is through an additive signal and noise model and an HMM state sequence $x(t)$ through a Markov chain that generates the baseline signal $s(t)$. Non-iterative and iterative cepstral compensation schemes have been derived in an attempt to equalize the source characteristics of the mismatched signal and what is effectively an N-vector word modelling the baseline signal (assuming signal representation by an N-state HMM). The latter is characterized by sample average of speech vectors in the HMM state and the average state segmentation points. Effectively, the E-step consists of the calculation of the length-normalized mean squared error between the two types of speech condition and the estimation of the state sequence. The M-step involves the calculation of the set of N state-based compensation vectors.

The problem of adapting the cepstral means to nominally arbitrary SNRs speech has not been dealt with comprehensively – the two levels of noise (low-noise and high-noise) have been the only preoccupation. Nevertheless, these two levels of noise are at the extreme of what would be expected in a real-life noisy car database and so constitute a good test. No prior knowledge of the noise cepstrum shape is explicitly used but is inferred from statistics of the different types of speech to reconstruct similar statistical distributions in the training and testing material.

Whole utterance averaging achieved by digit-dependent cepstral means collection is not sufficiently detailed and accurate compared to that achieved by digit-dependent, state-based cepstral means collection. The cepstral means adjustment on the training material gives better results than the cepstral means adjustment on the testing material. Noise immunity to low-noise speech (e.g. stationary noisy car environment) can be increased by cepstral means adjustment of the training material to resemble 'clean' speech together with cepstral means adjustment of the testing material given the *a priori* SNR. This breaks down for high-noise data. Application of SBCDC on the speech material has done better on the low-noise data (i.e. the higher SNR material) than on the high-noise data (i.e. the lower SNR material). The low-noise training data/high-noise testing case always gives better results than the clean training/high-noise testing case because the VQ codebook and subsequently the HMM contains some information about the engine noise.

Application of cepstral means compensation techniques when the identity of the speech token is the same as the reference model, whether iterative or non-iterative, can easily be shown to reduce the normalized MSE. Indeed, this procedure is commonplace in some speaker normalization algorithms or speech enhancement techniques. Difficulties arise for non-matching input token and reference model, and we have to rely on the modification of the non-matching token by SBCDC to yield a likelihood score worse than if the token and reference model match. The hypothesis-driven approach to applying SBCDC on the test material is not guaranteed to bring the acoustically compensated material to its correctly matched reference HMM all of the time.

The robustness of cepstral means adjustment to the imprecision associated with the state breakpoints is quite solid, as evidenced by the results obtained when *a priori* normalized duration points for a particular SNR is used for a different SNR. Detailed variation of the state-based cepstral means shift is necessary for proper reduction in the accumulated MSE but even gross variations achieved by extrapolation from data at another SNR do not affect the recognition accuracy too adversely.

Most of the experiments carried out could have been stopped at either just after the preprocessing or after the VQ encoding process, assuming that some means shift operation has been applied to the speech vectors. Indeed, the monitored MSE and the corresponding greatest reduction can provide a basis for recognition, as could be the minimum distortion recorded by the VQ process. However, the recognition process is carried one step further because of the benefits associated with implicit duration modelling in the Viterbi alignement which is different for each of the digits concerned. Nevertheless, from the present experiments, it has been found that the VQ distortion with beta distortion measure (and/or length-normalized MSE) and recognition error rate are in fact highly related.

Although SBCDC copes with some variations of the speaking rate and articulation by the time normalization, the method could be sensitive to unusually high or low speaking rate because of the greater variances associated with the state segmentation points and the collection of speech vectors in each state from the Viterbi alignemnt. Statistics of cepstral distributions retain the property of being invariant under linear distortions. However, there are many nonlinear distortions to the uncorrupted speech cepstral vector and, in extreme circumstances like the high-noise speech, SBCDC as currently carried out is not flexible enough.

It is hoped that this study contributes to the implementation of robust recognition systems that will continuously update the statistics of signal, noise, VQ codebook and HMM parameters for successful mismatched speech recognition caused by additive noise. The ultimate objective is for developing an algorithm that dynamically adapts to changes in acoustic environments, noise characteristics and level, introduction of new speakers and different microphones. It is also desirable to avoid the need, if possible, for collecting *a priori* statistics about these new conditions.

10.1 Future work

SBCDC can be adapted for other problems where there is a need to model mismatch statistics such as in speaker adaptation, codebook adaptation, stress compensation and speech enhancement. Conceptual problems remain to be studied before a full generalization of the method is possible. It is unclear whether state-based means shift is highly speaker-dependent, and if means shift is equally effective for delta-mfcc vectors and whether the beta distortion metric is applicable to delta-mfcc information. Second-order statistics have not been systematically collected and analysed because accurate estimation of the covariance matrices requires a larger database, and the process of estimating covariance matrices at various SNR would substancially increase the computational complexity required for the compensation process. SBCDC can also be modified in such a way that depends more on a localized SNR level. Real-time implementation problems need to be looked at more closely because of the additional memory storage and computation involved.

SBCDC needs modification for unlabelled continuous speech or unsegmented speech because of pauses which will result in the incorrect alignment of silence-type speech to the reference speech. To achieve correct alignments for input noisy connected or continuous speech, a silence HMM has to be inserted between each word HMM. In the stochastic segment model [44], a segment is transformed into a shorter resampled segment for known phoneme boundaries. For an unknown segmentation of phonemes, a Dynamic Programming algorithm is used with tentative re-sampling every other two frames to maximize the total likelihood score of the observation sequence. For a known digit identity, the utterance is effectively transformed by SBCDC into a 5-frame segment when using a 5-state HMM. The application of a future enhanced SBCDC to strings of noisy digits can follow a similar Dynamic Programming algorithm.

Multiple models per class are common to model in-class variability more accurately. Its usage is commonplace in statistical models, not least in hidden Markov modelling. This can be translated into a similar idea for SBCDC applied to words. The criteria for the calculation of multiple sets of state-based cepstral means shift per state for a particular word can be based upon the normalized state duration or the normalized Viterbi state likelihood score. In this way, multiple statistical means vectors in each state will characterize a particular word and provide more accurate in-class modelling.

11 Acknowledgements

The author is indebted to Trinity Hall for a scholarship. Part of the research has been carried out during an over-short stay at Ecole Nationale Supérieur de Telécommunications (ENST), made possible by Gerard Chollet and help provided by Chafik Mokbel, Francois Capman, Henri Barral, Francois Desbouvries and Philippe Dax.

References

[1] B. J. Stanton, L. H. Jamieson, and G. D. Allen. Acoustic-phonetic analysis of loud and lombard speech in simulated cockpit conditions. In *Proc. ICASSP*, pages 331–335, 1988.

[2] D. B. Pisoni et al. Some acoustic-phonetic correlates of speech produced in noise. In *Proc. ICASSP*, pages 1581–1585, 19.

[3] M. Berouti et al. Enhancement of speech corrupted by noise. In *Proc. ICASSP*, pages 208–211, 1979.

[4] W. M. Kushner et al. The effects of subtractive-type speech enhancement/noise reduction algorithms on paremeter estimation for improved recognition and coding in high-noise environments. In *Proc. ICASSP*, pages 211–215, 1989.

[5] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal Royal Statistics Soc. Ser. B (methodological)*, 39:1–38, 1977.

[6] R. A. Redner and H. F. Walker. Mixture densities, maximum likelihood and the EM algorithm. *Society For Industrial and Applied Mathematics Review*, 26(2):195–239, April 1984.

[7] L. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occuring in the statistical analysis of probabilistic functions of Markov chains. *Annals of Mathematical Statistics*, 41:164–171, 1970.

[8] L. E. Baum. An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes. *Inequalities*, 3:1–8, 1972.

[9] L. A. Liporace. Maximum likelihood estimation for multivariate observations of Markov sources. *IEEE Transactions on Information Theory*, IT-28:729–734, September 1982.

[10] E. L. Lehmann. *Theory of Point Estimation.* Wadsworth & Brookes/Cole. Statistics/Probability series, 1991.

[11] C. F. J. Wu. On the convergence properties of the EM algorithm. *The Annals of Statistics*, 11:95–103, 1983.

[12] Y. Ephraim et al. Speech enhancement based upon hidden Markov modelling. In *Proc. ICASSP*, pages 353–357, 1989.

[13] A. Dembo. Signal reconstruction from noisy partial information of its transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(1):65–72, January 1989.

[14] J. E. Porter and S. F. Boll. Optimal estimators for spectral restoration of noisy speech. In *Proc. ICASSP*, pages 18A.2.1–18A.2.4., 1984.

[15] G. Wong. *Improved Speech Hidden Markov Modelling Via An Expectation-Maximization Framework*. PhD thesis, Cambridge University, 1992.

[16] Y. Chen. Cepstral domain stress compensation for robust speech recognition. In *Proc. ICASSP*, pages 717–721, 1987.

[17] A. Acero and R. M. Stern. Environmental robustness in automatic speech recognition. In *Proc. ICASSP*, pages 849–853, 1990.

[18] M. Feng. Iterative normalization for speaker-adaptive training in continuous speech recognition. In *Proc. ICASSP*, pages 612–616, 1989.

[19] E. Bocchieri and G. R. Doddington. Frame-specific statistical features for speaker independent speech recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 34(4):755–764, August 1986.

[20] J. E. Shore and D. K. Burton. Discrete utterance speech recognition without time alignment. *IEEE Transactions on Information Theory*, IT-29:473–491, July 1983.

[21] B. H. Juang and L. R. Rabiner. Signal restoration by spectral mapping. In *Proc. ICASSP*, pages 2368–2372, 1987.

[22] A. Nadas, D. Nahamoo, and M. Picheny. Adaptive Labeling: Normalization of speech by adaptive transformations based on vector quantization. In *Proc. ICASSP*, pages 521–525, 1988.

[23] H. Gish, Y. Chow, and J. R. Rohlicek. Probabilistic vector mapping of noisy speech parameters for hidden Markov model word spotting. In *Proc. ICASSP*, pages 117–121, 1990.

[24] S. Roucos and A. M. Wilgus. Speaker normalization algorithms for very low rate speech coding. In *Proc. ICASSP*, pages 1.1.1–1.1.4., 1984.

[25] A. Nadas, D. Nahamoo, and M. A. Picheny. Speech recognition using noise-adaptive prototypes. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(10):1495–1503, October 1989.

[26] A. Varga et al. Noise compensation algorithms for use with HMM-based speech recognition. In *Proc. ICASSP*, pages 481–485, 1988.

[27] David B. Roe. Speech recognition using a noise-adapting codebook. In *Proc. ICASSP*, pages 1139–1143, 1987.

[28] S. Furui. Unsupervised speaker adaptation based on hierarchical spectral clustering. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(12):1923–1930, December 1989.

[29] R. P. Lippmann et al. Multistyle training for robust isolated-word speech recognition. In *Proc. ICASSP*, pages 705–708, 1987.

[30] J. H. L. Hansen and M. A. Clements. Stress compensation and noise reduction algorithms for robust speech recognition. In *Proc. ICASSP*, pages 266–270, 1989.

[31] S. B. Davis and P. Mermelstein. Comparison of parametric representation for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4):357–366, August 1980.

[32] A. Noll et al. Real-time connected word recognition in a noisy environment. In *Proc. ICASSP*, pages 679–681, 1989.

[33] Y. Linde, A. Buzo, and R. M. Gray. An algorithm for vector quantiser design. *IEEE Transactions on Communications*, 28:84–95, 1980.

[34] D. Mansour and B. H. Juang. A family of distortion measures based upon projection operation for robust speech recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(11):1659–1671, November 1989.

[35] L. R. Rabiner and B. H. Huang. An introduction to hidden Markov models. *IEEE ASSP Magazine*, pages 4–16, January 1986.

[36] Recognition rates obtained with noisy databases. Technical report of Esprit Project No. 2101 ARS, 1990.

[37] V. L. Beattie and S. J. Young. Hidden Markov model performance in noise. Technical Report CUED/F-INFENG/TR.49, Cambridge University Engineering Department, 1990.

[38] G. Wong. Cepstral means compensation for noisy speech recognition. ENST internal report, Jan. 1991.

[39] K. Shinoda, K. Iso, and T. Watanabe. Speaker adaptation for semi-syllable based continuous density hidden Markov model. In *Proc. ICASSP*, pages 857–861, 1991.

[40] D. H. Klatt. A digital filter bank for spectral matching. In *Proc. ICASSP*, pages 573–576, 1976.

[41] D. V. Compernolle. Noise adaptation in a HMM speech recognition system. *Computer Speech and Language*, 2:151–167, 1986.

[42] D. V. Compernolle. Increased noise immunity in large vocabulary speech recognition with the aid of spectral subtraction. In *Proc. ICASSP*, pages 1143–1147, 1987.

[43] Y. Ephraim, J. G. Wilpon, and L. R. Rabiner. A linear predictive front-end processor for speech recognition in noisy environments. In *Proc. ICASSP*, pages 1324–1328, 1987.

[44] M. Ostendorf and S. Roucos. A stochastic segment model for phoneme-based continuous speech recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(12):1857–1869, December 1989.