

Chapter 9. Robust Large Vocabulary Continuous Speech Recognition Based on Missing Feature Techniques

Yujun Wang, Maarten Van Segbroeck, Hugo Van hamme

Department ESAT/PSI, Katholieke Universiteit Leuven, Belgium

Abstract: Solutions for two important problems for the deployment of noise-robust large vocabulary automatic speech recognizers using the missing data paradigm are presented. A first problem is the generation of missing data masks. We propose and evaluate a method based on vector quantization and harmonicity that successfully exploits the characteristics of speech while requiring only weak assumptions on the noise. A second problem that is addressed is computational efficiency. We advocate the usage of PROSPECT features and the L -cluster- M -best method for Gaussian selection. In total, a speedup of a factor of about 6 can be achieved with these methods.

1. Introduction

In contrast to human listeners, automatic speech recognition (ASR) systems are particularly sensitive to the presence of background noises and acoustic variations in the speaking environment. Speech signals processed by an ASR-system are influenced by the speaker (e.g. gender, age, speaking style, emotion, accent, dialect, Lombard effect), by the surrounding sounds that add noise to the signal, by the microphone characteristics and by the transmission channel (i.e. the room impulse response). Robustness can be defined as the ability of the ASR to maintain its performance or degrade gracefully when exposed to a range of different conditions. This contribution deals only with robustness to noise. Channel effects, reverberation and speaker characteristics are not considered here.

The purpose of this contribution is not to review methods for automatic speech recognition in adverse environments. We will focus on the *missing data* or *missing feature* approach to robust speech recognition [1]. This method is motivated by analogy to human speech processing. Psycho-acoustic experiments have shown that speech contains sufficiently redundant information such that even when parts of the signal are completely removed, listeners are still capable of recognizing it by utilizing the information left in the distorted speech signal. In a spectrographic representation of noisy speech, some time-frequency regions will be dominated by the noise and others are dominated by the speech. Missing feature theory (MFT) attempts to compensate for additive noise distortions by first locating the corrupted time-frequency regions and then performing recognition on these partial or incomplete feature vectors. Therefore, MFT requires the estimation of a missing feature mask indicating the reliability of different spectral regions in the noisy data. Once the unreliable spectro-temporal regions are identified, further

processing requires only models of the speech signal. Hence, the noise characteristics only enter the mask estimation procedure. An important advantage of MFT is hence that different signal representations can be used to yield the spectro-temporal mask, a property that will be exploited in this contribution.

This text is organized as follows: in the next section, elements of a MDT-based automatic speech recognizer are introduced. In section 3, the evaluation tasks are described. Subsequently, we describe two core problems: mask estimation on real data in section 4 and numerical efficiency in section 5 and conclude in section 6.

2. Missing feature imputation for robust speech recognition

In MFT, recognition is based only on those regions in the time-frequency representation of the speech data that are not or mildly distorted by noise and hence well-matched with the recognizer's model. This requires a solution to the following problems: (i) locating of those matched regions in the time-frequency plane, (ii) a model evaluation method that handles the unreliable speech information, (iii) dealing with unreliable features in convolutional noise compensation. Therefore, three additions to the conventional architecture of the ASR are required, respectively a *Missing Feature Detector* (MFD), a speech reconstruction method which exploits the recognizer's acoustic model in the back-end and a MFT alternative for the commonly used cepstral mean normalization method. The last component is described in [22] and will not be discussed here. These modifications are schematically represented in Figure 1) with the boxes with dark background. The MDT can either make hard decisions about the reliability of spectro-temporal regions, or it can assign a probability of being reliable to each region. Though slightly better results have been reported with the latter approach ([17], [18], [20]), we will consider hard spectral masks

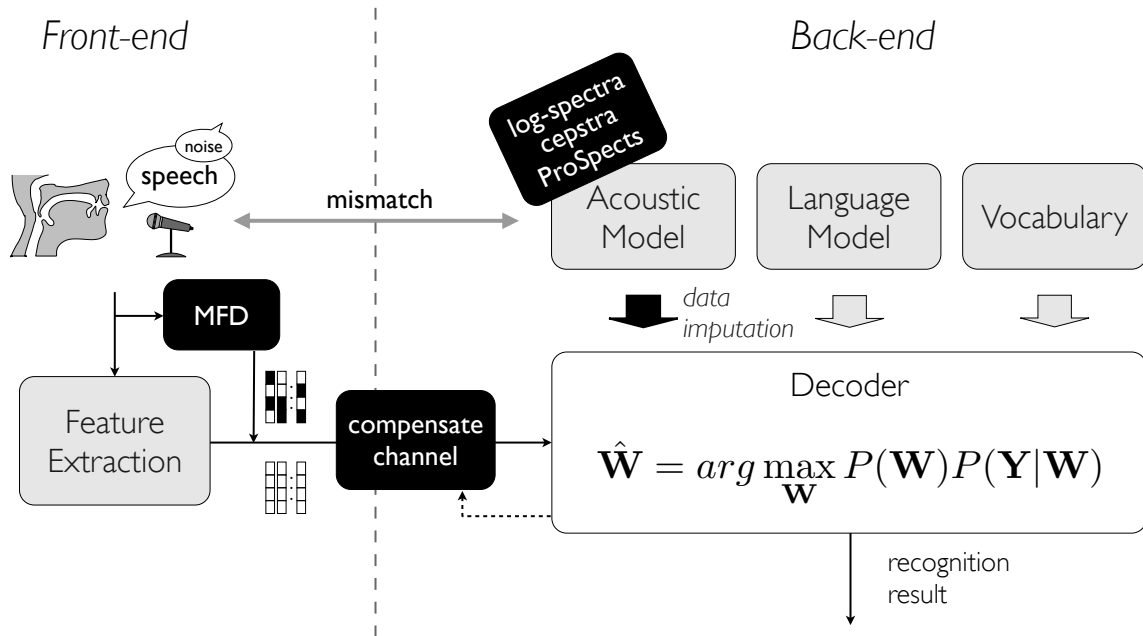


Figure 1: Schematic representation of the proposed MFT-based recognizer.

here. The rest of this section discusses the model evaluation with missing data.

In an HMM-based speech recognizer, the likelihood of the data is computed for each search hypothesis. When all data are observed, this likelihood is found by evaluating the statistical speech model at the observation. With missing data, we can choose to integrate the likelihood over the unknown observations, i.e. to marginalize out the missing data. Under suitable independence assumptions in the speech model, this leads to tractable computations and good robustness results [2]. However, many decades of speech research have shown these independence assumptions to be suboptimal and to lead to a loss in recognition accuracy. Without the independence assumptions, the integrals involved in marginalization do not have a closed form expression. Following [3], the integrals can be approximated, but it must be argued that there is no method to get arbitrarily close to the ideal result, i.e. the degree of approximation is not scalable.

In our work, we have chosen for an imputation approach in which the independence assumptions need not be made. In imputation, the missing data are first estimated and then substituted.

2.1 Masking

Referring back to Figure 1), the feature extraction module computes a sequence of D -dimensional vectors \mathbf{y} which contain the spectral log-energies observed in a filter bank with D channels. One such

vector will be called a *frame*. Though the filter bank outputs are sampled at regular instants (typically once every 10 ms), we will not explicitly annotate this time-dependency to simplify the notation. While \mathbf{y} denotes the response of the feature extraction module to the noisy signal, \mathbf{s} and \mathbf{n} will denote the output for the clean speech and the noise respectively. Assuming speech and noise are statistically independent, we can write:

$$\exp(\mathbf{y}) \approx \exp(\mathbf{s}) + \exp(\mathbf{n}) \quad (1)$$

where the $\exp(\cdot)$ function applies element-wise to vector components. Written as

$$\mathbf{y} \approx \log(\exp(\mathbf{s}) + \exp(\mathbf{n})) \quad (2)$$

the mutual masking effect of speech and noise becomes clear, since (2) can be approximated as $\mathbf{y} \approx \max(\mathbf{s}, \mathbf{n})$, where $\max(\mathbf{x}, \mathbf{y})$ of two D -dimensional vectors \mathbf{x} and \mathbf{y} denotes the D -dimensional vector obtained by taking the element-wise maximum. This also implies that \mathbf{y} serves as an upper bound for \mathbf{s} in its imputation.

In controlled experiments where noise is added to clean speech such as in the Aurora-4 database (see below), it is possible to identify the unreliable spectrographic areas from knowledge of the speech and noise:

$$\mathbf{s} \leq \mathbf{n} \quad (3)$$

We will refer to inequality (3) as the *oracle mask*.

2.2. Acoustic models

The acoustic model is a Hidden Markov Model (HMM) where the state-conditional emission densities are assumed to be described by a Gaussian Mixture Model (GMM). Hence, conditioned on HMM state q and

mixture component i , the speech is assumed to follow a multi-dimensional Gaussian distribution:

$$P(\mathbf{s} | q, i) = \frac{|\mathbf{P}|^{1/2}}{(2\pi)^{D/2}} \exp\left(-\frac{1}{2}(\mathbf{s} - \boldsymbol{\mu})' \mathbf{P}(\mathbf{s} - \boldsymbol{\mu})\right) \quad (4)$$

where $\boldsymbol{\mu}$ is the mean and \mathbf{P} is the precision (inverse covariance) matrix and $|\mathbf{P}|$ is the product of the \check{D} non-zero eigenvalues of \mathbf{P} . Equation (4) is a generic formulation which is also valid in case a singular or dimension-reducing transformation is applied to the spectra. For instance, it is quite common to transform log-spectra to cepstra:

$$\mathbf{c} = \mathbf{C}_K \mathbf{s} \quad (5)$$

where \mathbf{C}_K is a K -by- D Discrete Cosine Transform (DCT) matrix with $K \leq D$, which we will assume to have properly normalized rows such that it is orthonormal, i.e. $\mathbf{C}_K \mathbf{C}_K' = \mathbf{I}_K$. The Gaussian density is then

$$P(\mathbf{c} | q, i) = \frac{|\Sigma_c|^{-1/2}}{(2\pi)^{K/2}} \exp\left(-\frac{1}{2}(\mathbf{c} - \boldsymbol{\mu}_c)' \Sigma_c^{-1}(\mathbf{c} - \boldsymbol{\mu}_c)\right) \quad (6)$$

With $\check{D} = K$, $\mathbf{P} = \mathbf{C}_K' \Sigma_c^{-1} \mathbf{C}_K$ and $\boldsymbol{\mu} = \mathbf{C}_K' \boldsymbol{\mu}_c$, equation (6) reduces to (4). In case the spectral components are assumed to be independent (conditioned on the Gaussian), \mathbf{P} is a diagonal matrix.

2.3. Constrained optimization

The MDD produces a D -dimensional mask vector indicating which of the spectral observations in \mathbf{y} are reliable (subscript r) and which are unreliable (subscript u). Hence, we can conveniently write:

$$\mathbf{y}' = [\mathbf{y}'_u \quad \mathbf{y}'_r] \quad (7)$$

where apostrophe means vector or matrix transpose. We assume that the reliable observations \mathbf{y}_r follow the speech distribution. The unreliable components are assumed to be masked by noise, a stochastic process about which we do not wish to make strong assumptions regarding its distribution. Therefore, we will only impose the constraint that the speech which follows distribution (4), is bounded above by the observation:

$$\mathbf{s}_u \leq \mathbf{y}_u \quad \text{and} \quad \mathbf{s}_r = \mathbf{y}_r \quad (8)$$

The missing data will be imputed per Gaussian using the Maximum Likelihood Estimator (MLE), which is obtained by minimizing

$$(\mathbf{s} - \boldsymbol{\mu})' \mathbf{P}(\mathbf{s} - \boldsymbol{\mu}) \quad (9)$$

under the constraints (8).

2.4. Ternary masks

Most speech recognizers use derivative or delta features as well as the static features discussed above. Delta's are computed as linear combinations of feature values at successive frames. Because both positive and negative weights are used in the linear

combination, it is in general not possible to derive an informative upper or lower bound on the delta features to be imputed. One approach is to resort to *strict* masks [2], [19] where a feature is considered as unreliable as soon as one of the variables in the linear combination is unreliable. The clean value is then imputed without boundary constraints. This is a conservative approach. In practice, it is better to attempt to predict if a delta feature will be over or under estimated. To this end, consider the mask values per frequency bin as a sequence of ones (unreliable) and zeros (reliable). Now compute the delta of this sequence. If the result is zero, the delta feature is considered reliable and a zero mask value is assigned to this case. If it is positive (negative) the delta feature is considered as unreliable and imputed with the noisy delta feature as its upper (lower) bound and the assigned mask value is 1 (-1). In [24], we have shown experimentally that this is a workable approximation.

2.5. Non-negative least squares

To minimize (9) under equality and inequality constraints is called a constrained least squares problem. Define the diagonal D -by- D matrix $\boldsymbol{\Lambda}$ with the mask values along its diagonal and the slack variables \mathbf{x} , then

$$\mathbf{s} = \mathbf{y} - \boldsymbol{\Lambda} \mathbf{x} \quad \text{with} \quad \mathbf{x} \geq \mathbf{0}$$

The minimization of (9) then becomes:

$$\arg \min_{\mathbf{x} \geq \mathbf{0}} (\mathbf{y} - \boldsymbol{\Lambda} \mathbf{x} - \boldsymbol{\mu})' \mathbf{P} (\mathbf{y} - \boldsymbol{\Lambda} \mathbf{x} - \boldsymbol{\mu}) \quad (10)$$

which is a non-negative least squares problem (NNLSQ). Likelihood computation in a traditional HMM with emission densities described by Gaussian mixtures requires the evaluation of a quadratic, which requires about $3D$ multiplications per Gaussian. In contrast, in MDT, likelihood computation requires solving an NNLSQ problem. Solving this problem efficiently will be discussed below.

3. Recognizer and tasks

The methods presented in this work will be evaluated using an MDT extension of the large vocabulary continuous speech recognizer described in [21]. This system is particularly suited for our purpose since, compared to other large vocabulary systems with similar performance, it uses a relatively small number of Gaussians (typically 20,000 to 30,000). This is possible due to an advanced Gaussian tying scheme. The decoder requires a negligible amount of CPU resources in the MDT configuration and is based on token passing, supporting N -gram language models as well as context-free grammars. The front-end consists of a 22-channel MEL-scaled filter bank. The MDT-compatible channel compensation technique described in [22] is applied as well.

3.1. AURORA-4

A first test is based on the Aurora-4 large vocabulary database [36], which is derived from the WSJ0 Wall Street Journal 5k-word dictation task. Each test set consists of 330 utterances from multiple speakers, sampled at 16 kHz. The seven test sets have different noise types mixed in: no noise (01), car (02), babble (03), restaurant (04), street (05), airport (06), train (07) at an SNR-level that ranges from 5 dB to 15 dB. Test sets 01-07 were recorded with the same microphone as during training. The additional test sets (08-14) available in the database were recorded with a microphone selected from a set of different microphones. Since robustness to changes in channel is not our current focus, evaluation will be restricted to sets 01-07. The cross-word triphone acoustic model is trained on the clean training set yielding 4961 tied states with an average of 200 Gaussians per state (21037 Gaussians in total). The bigram language model for a 5k-word closed vocabulary is provided by Lincoln Laboratory.

3.2. SPEECHDAT-CAR

Since the Aurora-4 database contains artificially added noise, we also tested our approaches on the in-car data of the SpeechDat-Car Flemish database, which includes utterances recorded in different driving conditions. The signal of four microphones is simultaneously recorded at 16 kHz sampling rate. The microphones are placed at various distance from the speaker, from a close talking to a far field setting. The test task reported on here is the recognition of a command (mostly one word) out of a list of 600 possible commands. More tests on this database (not reported here) include the recognition of natural numbers, spelling, connected digit strings and dates/times.

The triphone acoustic model set is trained using the clean read speech component of the Flemish CGN database[23] and contains with 28917 tied Gaussians.

4. Mask estimation

A crucial part in a MFT-based recognizer is the computation of the reliability masks from noisy data. The reader is referred to [25] for an overview of methods. To accurately estimate masks in environments with unknown, non-stationary noise, only weak assumptions can be made about the noise, while we need to rely on a strong model for the speech. Knowledge about the human voice, such as spectral characteristics, harmonicity, energy patterns, voicing and onset characteristics should be exploited. It has been shown earlier that incorporating models of harmonicity of voiced speech is quite successful [26],

[27]. In this paragraph, we present a missing feature detector that uses harmonicity in the noisy input signal and a Vector Quantizer (VQ) to confine speech models to a subspace. As shown in [4], a constrained subspace for the spectral shape of speech signals can be captured in a vector quantization codebook trained on features extracted from clean speech. If speech is corrupted by noise, simple nearest-neighbor decoding fails due to the mismatch between the codebook training set and the noisy test set. Previous approaches for a more noise robust decoding were reported in [5], using a more perceptually motivated distance measure and [6], including phase derivatives.

Voiced speech is characterized by its strong harmonicity arising from the vocal chord vibration such that it can be decomposed into periodic (or harmonic) components at integer pitch multiples and the remaining aperiodic components. Therefore, we use the harmonic decomposition technique of [7], although alternative methods like comb filtering [8] could be used instead. This method will be restated in below and the harmonic decomposition masks derived from the decomposition signals will be defined as well. Then, we describe a VQ-based MFD that exploits the harmonicity of the speech by training the codebook on the spectral features extracted from the periodic and aperiodic part of the clean speech signal. During speech events, the decoding seeks to recover the original speech vector from the stored codewords by minimizing a cost function that can deal with additive noise corruptions. To compensate for linear channel distortions, the VQ-system self-adjusts its codebook to the channel during online recognition.

4.1. Harmonic decomposition

To decompose a noisy speech signal into a periodic and an aperiodic component, a pitch estimate is first computed by a subharmonic summation method [10]. The signal is then subsequently framed in overlapping *segments* with a length of two pitch periods and a single pitch period of frame-shift. If p is the pitch epoch index and Ω_p the estimate of the double pitch period, then the noisy speech signal is written as:

$$y_p(n) = h_p(n) + r_p(n) \quad \text{with } 0 \leq n < \Omega_p \quad (11)$$

where $h_p(n)$ is the periodic or harmonic signal component and $r_p(n)$ is the residual, which will be also referred to as the aperiodic signal component. The periodic signal component is described by the model:

$$h_p(n) = \left(1 + \frac{c_p n}{\Omega_p} \right) \left[\sum_{k=0}^{K_p} a_{k,p} \cos(2\pi f_{0,p} kn) + \sum_{k=1}^{K_p} b_{k,p} \sin(2\pi f_{0,p} kn) \right] \quad (12)$$

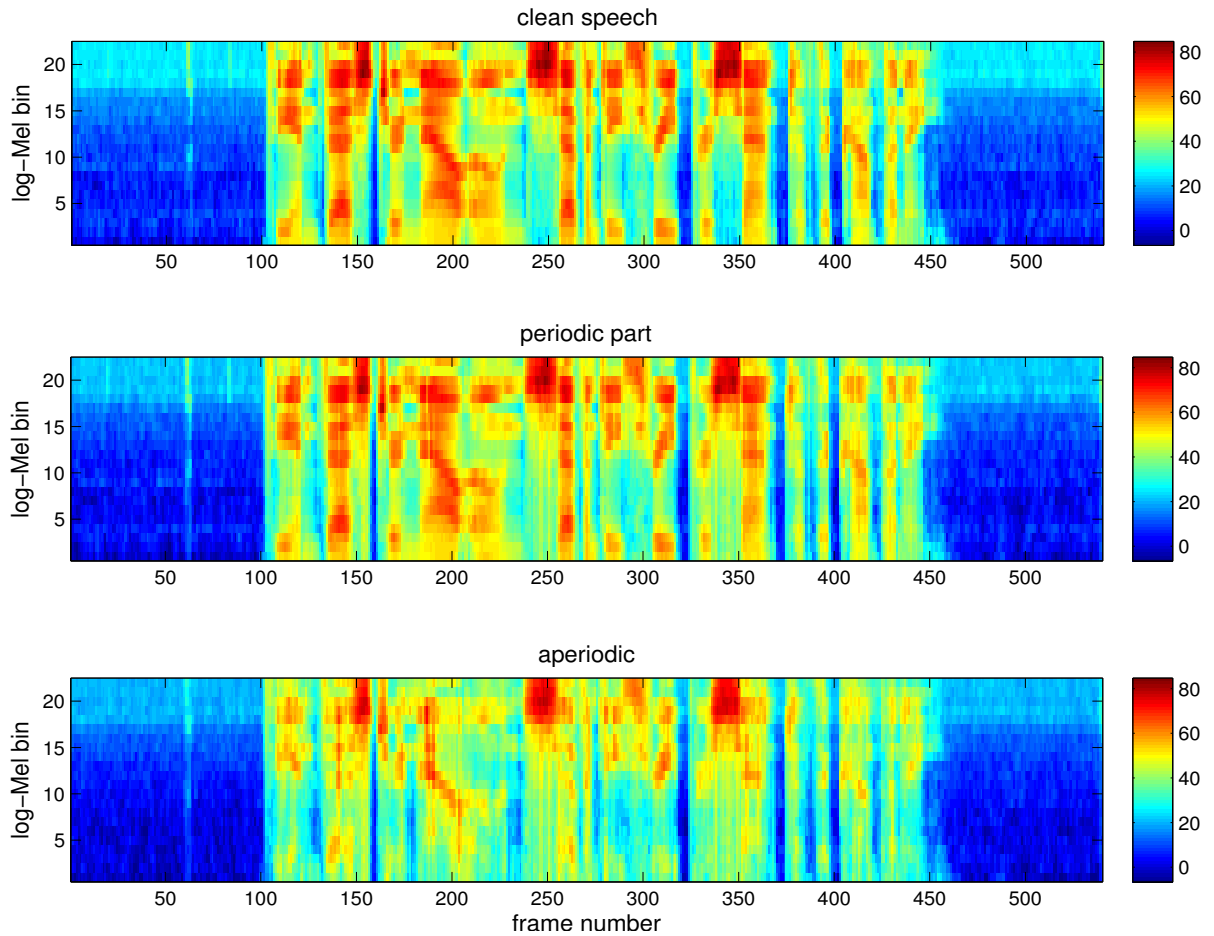


Figure 2: Log-Mel spectrogram of the clean utterance (top) and its corresponding periodic (middle) and aperiodic (bottom) part obtained after harmonic decomposition

where K_p is the number of harmonics that needs to be considered up to the Nyquist frequency in the p -th pitch epoch, i.e. K_p is the largest integer such that $K_p f_{0,p} < 0.5$. The modulation parameter c_p describes a common linear amplitude trend of all harmonics. The parameters in (12) are estimated in the least squares sense. For each choice of c_p and $f_{0,p}$ the estimation of the amplitudes is a linear least squares problem which has an analytic solution and can be substituted back in the least squares cost function. What remains is a cost function in c_p and $f_{0,p}$ which is minimized with a Levenberg-Marquardt quadratic programming technique [11].

The per-epoch decompositions (11) and (12) are combined with overlap-add to form contiguous periodic and aperiodic signal components and subsequently processed with the same MEL-scale analysis as the noisy signal to obtain the per-frame log-spectral periodic (\mathbf{h}) and aperiodic (\mathbf{r}) feature vectors. Hence:

$$\exp(\mathbf{y}) \approx \exp(\mathbf{h}) + \exp(\mathbf{r}) \quad (13)$$

The log-Mel representations of the periodic and aperiodic part are shown in Figure 2) for a clean

utterance from the Aurora-4 database “*The investor now owns seventy three percent of the company*”. During voiced speech, the periodic component is stronger than the aperiodic component. During unvoiced speech, both components are of equal strength. This can be understood as follows. Consider a voiced speech signal. Given that the harmonic decomposition (11) and (12) is valid over the double pitch period, the amplitudes $a_{k,p}$ and $b_{k,p}$ could be found as the real and imaginary part of the even-numbered spectral lines of the discrete Fourier transform of the signal (at least if c_p is zero and Ω_p is an integer number). These spectral lines will contribute to \mathbf{h} . The interleaved spectral lines make up the residual \mathbf{r} . A typical spectrum is shown in the left panel of Figure 1). For unvoiced speech, the pitch estimator will produce an arbitrary value, but the decomposition (11) and (12) can still be computed. A typical spectrum is given in the right panel of Figure 3). As before, by construction, the even spectral lines contribute to \mathbf{h} and the odd ones to \mathbf{r} . If the spectrum is smooth, \mathbf{h} and \mathbf{r} will be nearly equal, which is observed in Figure 2). The formulation of the signal decomposition as a parameter estimation problem

extends the motivation given above to cases where $1/f_{o,p}$ is not an integer and c_p is nonzero. Below, we will describe and compare two methods for mask generation that exploit this harmonic decomposition.

4.2. Harmonicity masks

The main assumption made in this method is that the noise has a smooth spectrum and hence does not contain strong harmonic components. The mask estimation method is based on a model that estimates clean speech from the periodic component of the noisy signal by applying a time (frame) and frequency-dependent gain γ :

$$\exp(\mathbf{s}) = \gamma \odot \exp(\mathbf{h}) \quad (14)$$

where \odot denotes element-wise product of vectors. With (1), (13) and (14) the masking decision criterion (3) can be rewritten as:

$$(2\gamma - \mathbf{1}) \odot \exp(\mathbf{h}) \geq \exp(\mathbf{r}) \quad (15)$$

where $\mathbf{1}$ is a D -dimensional vector of ones. What remains to be done is to establish a method of estimating γ . For noisy speech, the noise will add to all spectral lines in **Figure 3**, so the periodic component \mathbf{h} of the noisy signal is composed of the periodic component of the clean speech plus the spectrum of the noise at the multiples of the pitch frequency. Hence, the noise adds to the periodic components and the more noise, the smaller γ needs to be. With the assumed smoothness of the noise, the noise level at the even numbered spectral lines can be estimated from the odd-numbered spectral lines, i.e. the aperiodic signal part, as illustrated in the right panel of **Figure 3**, an idea that relates to harmonic tunneling [13]. Since the aperiodic signal part is not affected by the strong periodic components stemming from voiced speech, it is easily tracked with the minimum statistics technique [14]. This leads to the following gain estimator:

$$2\gamma - \mathbf{1} \approx \mathbf{1} - 2\zeta \frac{\bar{\mathbf{r}}}{\mathbf{h}}$$

where the division is again element-wise, $\bar{\mathbf{r}}$ is the

minimum statistic of the sequence \mathbf{r} , i.e. the element-wise minimum of \mathbf{r} over sliding window of frames and ζ is the correction factor to map the minimum statistics to instantaneous spectral estimates.

4.3. Vector quantization masks

The algorithm for estimating VQ-masks presented below is based on a source separation technique. The strategy is to estimate the unknown clean speech and noise from the noisy observation. The missing feature mask will then be constructed by comparing both estimates in the time-frequency plane. In the literature, the detection of missing data often relies on important assumptions about the noise type. In environments that do not meet these assumptions, the accuracy of the mask will be disappointing. Therefore, more constraints should be placed on the speech. To this end, we make use of a VQ-based approach with a codebook representing a constrained space for the spectral shape (timbre) of human speech.

In vector quantization, the best codebook entry to represent the observed feature vector is selected by some distance measured between the noisy speech and the codebook vectors. Since it is the intention of the MFT-approach to avoid multi-style training, the codebook is trained on clean speech data. As a consequence, the decoding will be less accurate, especially at low SNRs. As will be explained next, the VQ-based mask estimator is made more noise robust by choosing an appropriate codebook and incorporating noise compensation steps in the decoding process.

4.3.1. Codebook

A codebook trained on clean speech will mismatch the noisy input speech. We try to compensate for this mismatch by exploiting the relation between the periodic and aperiodic part of speech. The harmonic decomposition method presented above is therefore applied to the clean speech training set. Subsequently, the log-Mel scaled features of the periodic and aperiodic part are stacked into a single vector of dimension $2D$. By training a codebook on these data, it will capture not

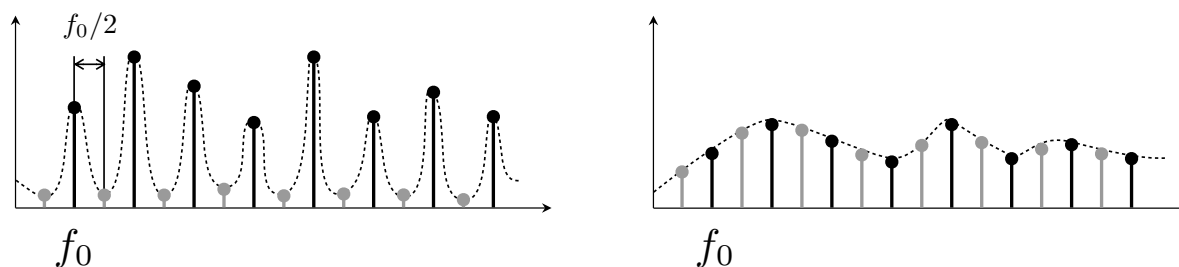


Figure 3: Harmonic decomposition of voiced speech (left) and noise without harmonicity structure (right).

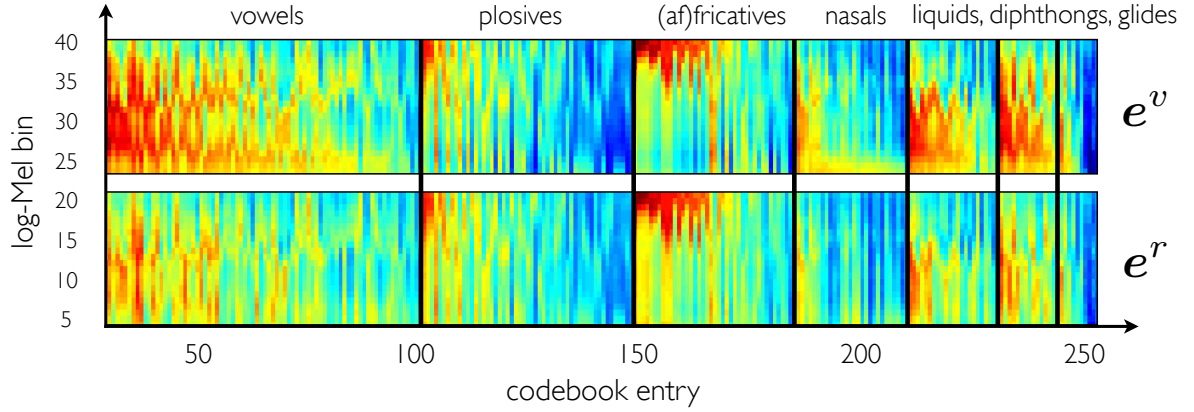


Figure 4: VQ-codebook with 250 entries trained on the periodic and aperiodic parts of the clean Aurora-4 data and representing different broad phonetic classes

only the spectral shape of the speech but also the relation between the spectral shape of the periodic and aperiodic speech components. Here, we assume that in testing conditions the noise is not harmonic or, less stringent, that the energy of the speech harmonics dominates over that of the noise such that the harmonic decomposition method finds the correct speech pitch and thereby the speech harmonics. Imposing constraints on the range and the continuity of the pitch as in [12] could further improve the robustness to noise and in particular to harmonic structure in the noise.

Training is performed by the k-means algorithm initialized by randomly selected data vectors. Frames corresponding to silence will be disregarded for reasons which become clear in the next subsection. In an additional refinement step, the codebook outliers are replaced by performing 2-means clustering onto the clusters with the highest variance. The l -th codebook entry will be denoted by

$$\mathbf{e}_l = \begin{bmatrix} \mathbf{e}_l^{(h)} \\ \mathbf{e}_l^{(r)} \end{bmatrix}$$

where the superscripts indicate the periodic and aperiodic parts.

Large vocabulary recognition tasks require a codebook that provides a set of codewords representative for all phones. Therefore, the clean training data is firstly categorized into broad phonetic classes: vowels, diphthongs, glides, liquids, fricatives plus affricates, nasals and stops. A sub-codebook is then trained for each phonetic class. The number of codewords per sub-codebook is proportional to the prior probability of occurrence of the phonetic class it represents. Table 1 shows these percentages for the training set of the Aurora-4 database. The corresponding 250-entry codebook is illustrated in Figure 4). Note that the periodic part of vowels is more energetic than the aperiodic part, while for e.g. affricatives the relationship between both parts is a similar energy contour.

Table 1. Prior probabilities in % of broad phonetic classes in the training database Aurora-4.

vowel	diph- thong	plosive	fricative	affricate	nasal	glide	liquid
32.3	6.0	21.2	16.5	1.1	11.3	2.4	9.2

4.3.2. Vector quantization of noisy speech

Since the codebook only represents a model for the human voice, decoding in non-speech (or noise) frames will lead to incorrect codebook matching and misclassifications in the mask. Therefore, a *Voice Activity Detector* (VAD) based on the integrated bi-spectrum, inspired by [15], will segment speech from non-speech frames in order to restrict the decoding to speech events. For a frame labeled as non-speech, all mask values will be set to one, indicating that all components are unreliable.

After construction of the per-frame periodic and aperiodic spectral feature vectors \mathbf{h} and \mathbf{r} , a search is done through the entire codebook to find the best matching codeword by minimizing some distance metric between these vectors and the codewords:

$$\min_l \mathcal{D}([\mathbf{h}' \ \mathbf{r}'], [\mathbf{e}_l^{(h)'} \ \mathbf{e}_l^{(r)'}])$$

Conventionally, the distance metric would be a vector norm between the data and the codewords. If noise is added to the speech, \mathbf{h} and \mathbf{r} will also contain noise components, which will perturb the vector quantization process. We therefore formulate the VQ distance metric as follows [9]:

$$\begin{aligned} \mathcal{D}([\mathbf{h}' \ \mathbf{r}'], [\mathbf{e}_l^{(h)'} \ \mathbf{e}_l^{(r)'}]) \\ = \min_{\mathbf{u}, \mathbf{v}} \|\mathbf{h} - \max(\mathbf{e}_l^{(h)}, \mathbf{u})\|^2 + \|\mathbf{r} - \max(\mathbf{e}_l^{(r)}, \mathbf{v})\|^2 \\ + \|\mathbf{u} - \mathbf{v}\|^2 + \|\mathbf{v} - \bar{\mathbf{r}}\|^2 \end{aligned} \quad (16)$$

Equation (16) is motivated as follows: in the first term, the max-operator combines the periodic component of

VQ-constrained speech estimate $\mathbf{e}_l^{(h)}$ and of the noise estimate \mathbf{u} to a periodic component estimate which should match the observed noisy periodic component estimate \mathbf{h} as closely as possible. The second term does the same for the aperiodic component. The third term expresses that the periodic and aperiodic components of the noise estimate should resemble, i.e. that the noise should not have a strong harmonic structure. The fourth term expresses that the instantaneous aperiodic noise estimate should be close to a long-term minimum of \mathbf{r} , denoted by $\bar{\mathbf{r}}$. Again, the reasoning is that the aperiodic signal component is not as much affected by the harmonic structure of speech and therefore the minimum statistics approach [14] will work better than on undecomposed spectra.

Given the codeword index l , the minimization over \mathbf{u} and \mathbf{v} in equation (16) decomposes into D independent two dimensional optimization problems which are easily solved [9]. Once the speech and noise estimates are known, a mask is easily derived by comparing both components. An example is shown in Figure 6).

To improve the coherence of the codeword sequence, extra temporal constraints could be taken into account by training the codeword transitions in a bigram model on top of the VQ. Instead of per-frame VQ, a Viterbi search algorithm is now required to find the optimal label sequence over the complete utterance. However, experiments have shown that this does not lead to an increase in performance of our system. This can be attributed to the fact that local errors due to approximations in the model tend to propagate over multiple frames. As a consequence, the bigram will substitute the well-matching codewords by incorrect ones. This effect has also

been discussed in [6]. Note that the use of temporal dependencies masking decisions has also been exploited in the approach of [16].

4.4. Evaluation

By training the codebook on different phoneme categories as explained above, a codebook of 500 entries suffices to capture the most important spectral variations of the large vocabulary Aurora-4 database. Figure 5) presents the word error rate for the MFT-based recognizer using oracle (OR), harmonic decomposition (HD) and VQ-masks. As can be seen from the results, the VQ-based masks are significantly more accurate than the HD-masks for all the test sets. For HD-masks, the decision criterion uses the idea that the periodic part will be dominated by the speech. This often leads to poor decisions in unvoiced speech segments. In the VQ-based approach however, the spectral shape constraints expressed by the codebook will provide important masking information for unvoiced speech fragments.

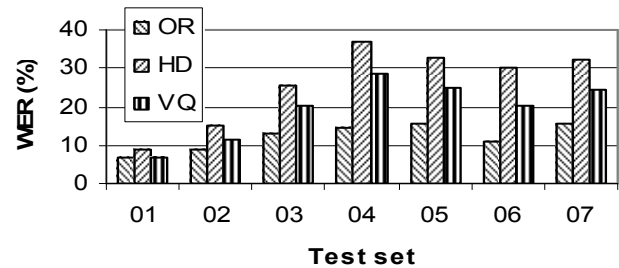


Figure 5: Word error rate (in %) on the Aurora-4 test sets with close-talk microphone using MFT with oracle (OR), harmonic decomposition (HD) and VQ-masks.

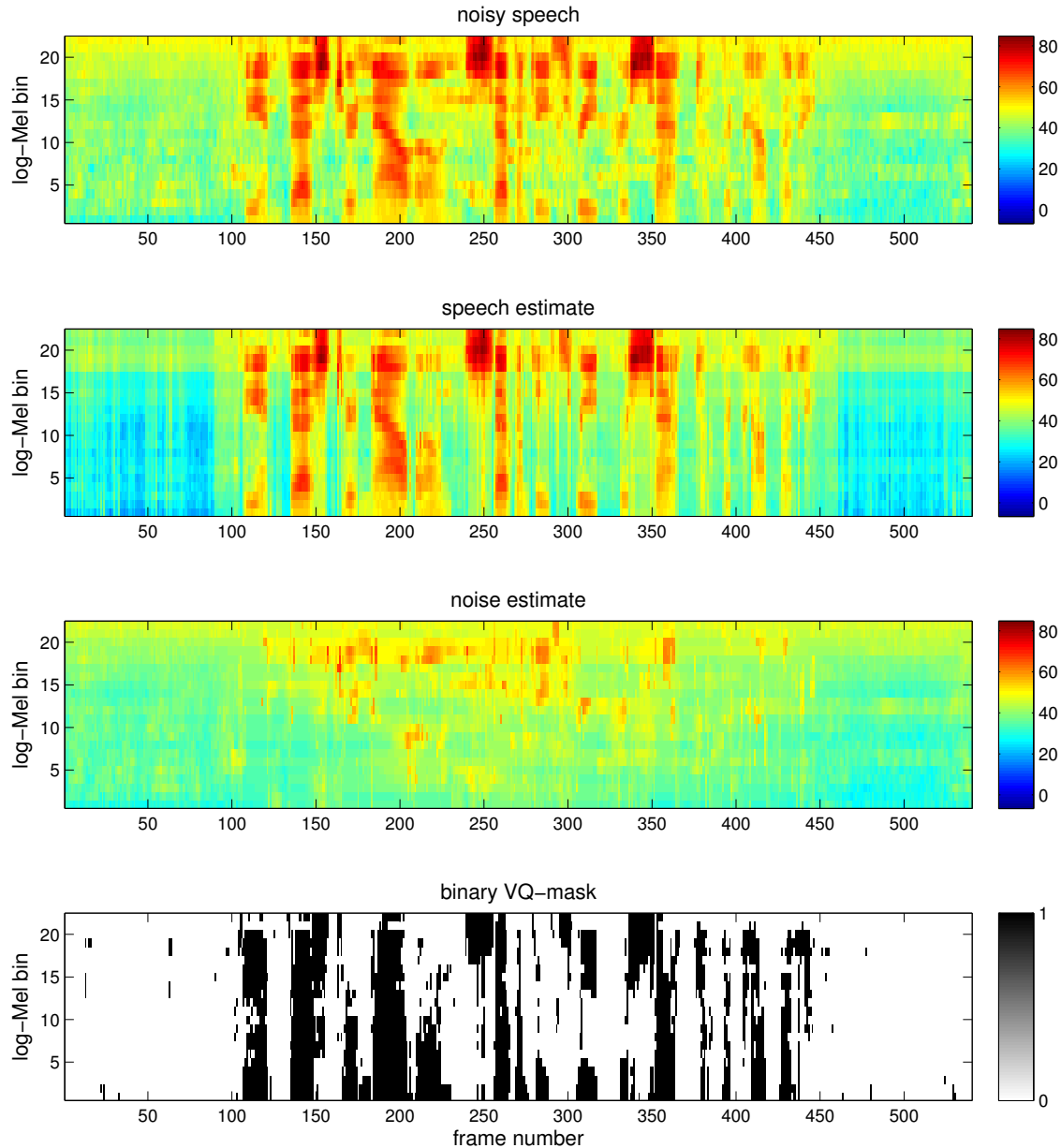


Figure 6: Input noisy speech (top), VQ-constrained speech estimate, noise estimate and VQ-mask (bottom) of the noisy utterance “*The investor now owns seventy three percent of the company*” of the Aurora-4 database corrupted by restaurant noise at 10 dB SNR. The dark areas are reliable. In the top three panels the color bar is expressed in dB.

5. Numerical efficiency

In (10), the imputation of missing data was formulated as a NNLSQ problem. The minimal value of the cost function also yields the log-likelihood, which is ultimately the quantity that is needed for speech recognition. Since \mathbf{P} in (9) is in general not diagonal, solving this problem is expensive compared to the evaluation of a quadratic in conventional HMM-based speech recognizers with GMMs as emission densities. In this section, we discuss methods to efficiently solve the NNLSQ problem.

First, we will introduce a feature representation which will make the gradient computation more efficient in a gradient descent (GD) approach. Then, we discuss and evaluate methods for Gaussian selection.

5.1. PROSPECT features and gradient descent

PROSPECT features [28] aim to reduce the computational load in MDT by working with a linear transforms that can be factorized with matrices of small size. The main idea is that the main correlations in log-spectral features can be described by a low order cepstrum. The

correlations in the residual (i.e. the spectrum left after removing the contribution attributed to the low order cepstrum) are then disregarded, i.e. the residual is modelled by a Gaussian with a diagonal covariance. The transformation applied to the log-spectrum \mathbf{s} is then:

$$\mathbf{p} = \begin{bmatrix} \mathbf{c} \\ \mathbf{d} \end{bmatrix} = \begin{bmatrix} \mathbf{C}_K \\ \mathbf{I}_K - \mathbf{C}_K' \mathbf{C}_K \end{bmatrix} \mathbf{s} = \begin{bmatrix} \mathbf{C}_K \\ \mathbf{P}_K^\perp \end{bmatrix} \mathbf{s}$$

i.e. the $K+D$ -dimensional PROSPECT feature vector \mathbf{p} is composed of a low-order cepstrum \mathbf{c} and a D -dimensional residual $\mathbf{d} = \mathbf{s} - \mathbf{C}_K' \mathbf{c}$ obtained by subtracting the spectral contribution of \mathbf{c} . Algebraically, the residual is the projection of \mathbf{s} on the orthogonal complement of the row-space of \mathbf{C}_K .

Experiments on large vocabulary tasks have shown that even with an order K as low as 3 or 4, a system using PROSPECT features or cepstral features obtains comparable accuracy when using the same number of Gaussians with diagonal covariance [28]. It is easy to see that \mathbf{d} actually contains correlated features, since it is obtained by projecting the random variable \mathbf{s} on a subspace of lower dimension. To compensate for this, a constant stream weight α [29] is applied:

$$P(\mathbf{s} | q, i) = N_c N_d^\alpha \quad (17)$$

with

$$N_c = \frac{1}{(2\pi)^{K/2} |\Sigma_c|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{c} - \boldsymbol{\mu}_c)' \Sigma_c^{-1} (\mathbf{c} - \boldsymbol{\mu}_c)\right)$$

$$N_d = \frac{1}{(2\pi)^{D/2} |\Sigma_d|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{d} - \boldsymbol{\mu}_d)' \Sigma_d^{-1} (\mathbf{d} - \boldsymbol{\mu}_d)\right)$$

where Σ_c and Σ_d are diagonal. With this model choice, we can derive:

$$\mathbf{P} = \mathbf{C}_K' \Sigma_c^{-1} \mathbf{C}_K + \alpha \mathbf{P}_K^\perp \Sigma_d^{-1} \mathbf{P}_K^\perp \quad (18)$$

For solving the NNLSQ problem, we use a gradient descent algorithm. At each iteration step, the computation of the gradient involves multiplication with \mathbf{P} , which thanks to (18) involves only matrix multiplications of size at most K -by- D , which makes PROSPECT features more efficient than cepstral features.

5.2. Gaussian selection

Apart from optimizing the evaluation of the Gaussians in an MDT context, we can also try to avoid the evaluation using Gaussian selection. The motivation is that Gaussians with a mean a few standard deviations away from the (clean) feature vector have a close-to-zero likelihood and hardly contribute to the optimal decoding path, i.e. the feature vector lies in the tail of these Gaussians. Though methods for Gaussian selection share a common procedure, quantization or clustering in the training phase and selection in the decoding phase, they can be classified as cluster-based or scalar

methods. Scalar methods such as Bucket Box Intersection (BBI) [30] or the Fast Removal of Gaussians (FRoG) [31] quantize the observation space using binary decision trees involving only one feature dimension per decision. FRoG sets up binary decision trees for every axis of the observation space and assigns a Gaussian removal list for every node of the tree. The Gaussians remain to be evaluated are removed from a list while the tree is traversed during decoding. BBI sets up a binary KD-tree to quantize the observation space. Questions are asked dimension per dimension to reach a leaf node which holds a list of selected Gaussians. In an MDT method, the feature vector values are only known after solving the NNLSQ problem, such that scalar methods become inefficient when applied to noisy feature vectors. Clustering methods such as [33], [34] and [35] cluster all the Gaussians in the training phase. During decoding, the cluster Gaussians are evaluated first and subsequently the member Gaussians attached to the clusters are evaluated. Since the cluster Gaussians can be evaluated with MDT, cluster methods are suitable for our application. In [34], a neighborhood is created based on each cluster Gaussian to include more member Gaussians around the clusters. During decoding, the neighborhood of the cluster that has the best likelihood is selected. Methods [33] and [35] select multiple best clusters during decoding. The number of selected clusters in [33] is fixed, while in [35] it varies frame by frame and is controlled by the likelihood of the most unlikely clusters. Here, two variants of cluster-based methods are compared: an L -cluster- M -best scheme [33] and a Neighborhood method based on [34]. Finally, the computational cost can also be controlled by reducing the number of Gaussians without affecting the model accuracy. In subspace clustering [38], computational and memory savings can be achieved by dividing the whole feature space into several small streams that can each be modeled by a smaller number of Gaussians. The Gaussian parameters in the full acoustic space are the concatenation of the stream Gaussians. In MDT-systems, the imputation is done per feature stream (i.e. static and delta streams). A choice of streams smaller than the full spectrum or its delta's would weaken the speech model for imputation. We will therefore restrain our analysis to subspaces that coincide with the different delta streams using various cluster sizes.

Below, we will first specify the distance metric used to cluster Gaussians in the PROSPECT domain with the k-means algorithm. Then, we describe how the cluster centroids are computed from the member Gaussians. Finally, we provide the details of both cluster-based Gaussian selection methods.

5.2.1. Gaussian clustering

The symmetric Kullback-Leibler Divergence (KLD) is used as a distance metric in k-means clustering of N -dimensional Gaussians. The KLD between two

Gaussians f and g with diagonal covariance matrix is [32]:

$$\begin{aligned} d(f, g) &= \text{KLD}(f \parallel g) + \text{KLD}(g \parallel f) \\ &= \int f \log \frac{f}{g} dx + \int g \log \frac{g}{f} dx \\ &= \frac{1}{2} \sum_{i=1}^N \left(\frac{\sigma_{gi}^2}{\sigma_{fi}^2} + \frac{\sigma_{fi}^2}{\sigma_{gi}^2} + \frac{(\mu_{gi} - \mu_{fi})^2}{\sigma_{fi}^2} + \frac{(\mu_{gi} - \mu_{fi})^2}{\sigma_{gi}^2} \right) - N \end{aligned}$$

However, some care must be taken: $P(\mathbf{s}|q, i)$ formulated by equation (17) must be renormalized to make sure it integrates to unity:

$$\begin{aligned} P(\mathbf{s} | q, i) &= \frac{1}{(2\pi)^{\frac{K+D}{2}} \prod_{k=1}^K \sigma_{ck} \prod_{j=1}^D \sqrt{\alpha}} \times \\ &\exp \left(-\frac{1}{2} \sum_{k=1}^K \frac{(c_k - \mu_{ck})^2}{\sigma_{ck}^2} - \frac{1}{2} \sum_{j=1}^D \frac{(d_j - \mu_{dj})^2}{(\sigma_{dj} / \sqrt{\alpha})^2} \right) \end{aligned} \quad (19)$$

The divergence hence becomes:

$$d(f, g) = d(f_c, g_c) + \alpha d(f_d, g_d) + A - (1 - \alpha)D \quad (20)$$

with

$$\begin{aligned} d(f_c, g_c) &= \frac{1}{2} \sum_{k=1}^K \left(\frac{\sigma_{gck}^2}{\sigma_{fck}^2} + \frac{\sigma_{fck}^2}{\sigma_{gck}^2} + \frac{(\mu_{gck} - \mu_{fck})^2}{\sigma_{fck}^2} + \frac{(\mu_{gck} - \mu_{fck})^2}{\sigma_{gck}^2} \right) - K \\ d(f_d, g_d) &= \frac{1}{2} \sum_{j=1}^D \left(\frac{\sigma_{gdj}^2}{\sigma_{fdj}^2} + \frac{\sigma_{fdj}^2}{\sigma_{gdj}^2} + \frac{(\mu_{gdj} - \mu_{fdj})^2}{\sigma_{fdj}^2} + \frac{(\mu_{gdj} - \mu_{fdj})^2}{\sigma_{gdj}^2} \right) - D \\ \text{and} \\ A &= \frac{1}{2} \sum_{j=1}^D \left(\frac{(1 - \alpha)\sigma_{gdj}^2}{\sigma_{fdj}^2} + \frac{(1 - \alpha)\sigma_{fdj}^2}{\sigma_{gdj}^2} \right) \end{aligned} \quad (21)$$

μ_{fck} and σ_{fck} are the k -th component of the mean and diagonal covariance of the cepstral part of distribution f ; μ_{gck} and σ_{gck} are the counterparts of g . μ_{fdj} and σ_{fdj} are j -th component of the mean and diagonal covariance of the residual part of f ; μ_{gdj} and σ_{gdj} are the counterparts of g . We have observed that omitting A from equation (20) leads to a better balancing of cluster sizes and a better computation/accuracy trade-off. When computing the distance between a Gaussian (smaller variance) and a cluster candidate (larger variance) we observe from (21) that due to A , a cluster with a large variance in its projection part may be disfavoured. Hence the metric becomes

$$d(f, g) = d(f_c, g_c) + \alpha d(f_d, g_d)$$

5.2.2. Parameter Estimation of Cluster Gaussians

Like in [32], the cluster centre is chosen by unweighted matching of the first and the second order moments with the clustered Gaussians. Hence, its mean and diagonal covariance are given by:

$$\bar{\mu}_i = \frac{1}{W} \sum_{k=1}^W \mu_{ki} \quad (22)$$

$$\bar{\sigma}_i^2 = \frac{1}{W} \sum_{k=1}^W (\sigma_{ki}^2 + \mu_{ki}^2) - \bar{\mu}_i^2 \quad (23)$$

where W is the number of Gaussians belonging to the specific cluster. μ_{ki} and σ_{ki} are the i -th component of mean and diagonal covariance of k -th member Gaussian. Finally, the PROSPECT means in (22) are transformed to the spectral domain as

$$\bar{\boldsymbol{\mu}}_s = \bar{\boldsymbol{\mu}}_d + \mathbf{C}_K' \bar{\boldsymbol{\mu}}_c \quad (24)$$

with $\bar{\boldsymbol{\mu}}_c = [\bar{\mu}_1 \dots \bar{\mu}_K]'$ and $\bar{\boldsymbol{\mu}}_d = [\bar{\mu}_{K+1} \dots \bar{\mu}_{K+D}]'$. With variances (23) substituted in (18) and mean (24) substituted in (9), the cluster Gaussians can be evaluated using MDT.

5.2.3. L -cluster- M -best method

In this method, the Gaussians are clustered off-line into L clusters. Given a noisy observation \mathbf{y} , each of the L cluster Gaussians is evaluated using MDT and all members of the M best-scoring cluster Gaussians are subsequently evaluated using MDT.

5.2.4. Neighborhood method

In this method, the Gaussians are clustered into L clusters. Given a noisy observation \mathbf{y} , the best scoring cluster Gaussian j is identified. Subsequently, all Gaussians satisfying

$$d(\text{gaussian}, \text{cluster}_j) < \theta \frac{1}{N_j} \sum_{k=1}^{N_j} d(\text{member}_k, \text{cluster}_j)$$

are evaluated using MDT, where θ controls the neighborhood size and N_j is the number of member Gaussians of cluster j . Notice that the list of Gaussians satisfying this criterion can be precomputed off-line.

5.3. Initialization

With gradient descent and given that cost function (9) is convex, a sufficiently large number of iterations will always bring the estimate arbitrarily close to the optimal solution of the NNLSQ problem. The number of iterations required to approach the optimal solution such that no accuracy loss is observed, is however affected by the choice of the initial point. The initial point must be feasible (i.e. it must satisfy the non-negativity constraint).

To initialize the NNLSQ problem for the cluster Gaussians, we maintain only the diagonal of \mathbf{P} . In this case, the problem reduces to D independent scalar problems that are easy to solve and involve only a max-operation [1]. For a member Gaussian, the MDT solution of its cluster Gaussian appears to be a sufficiently good initialization such that the accuracy does not improve beyond one or two GD iterations and hence, from a practical point of view, the optimization algorithm has converged.

5.4. Evaluation

Both the L -cluster- M -best and the neighbourhood methods of Gaussian selection for PROSPECT MDT

were implemented and tested in our experiments, as well as the subspace method. The experiments are carried out on the car test set (noise type 02) of the Aurora-4 [36] database. Since the noise in Aurora-4 is artificially added, we also evaluate on the SpeechDat [37] in-car Flemish database. The SNR of all utterances was measured, such that an analysis of word error rate (WER) versus SNR becomes possible.

5.4.1. Experiments on Aurora-4

An acoustic model in the PROSPECT domain with $K=4$ and $D=22$ is derived for the 21037 Gaussians (see section 3; **Error! No se encuentra el origen de la referencia.**) by single pass retraining on the clean training data set. A VQ mask as described in section 4 is computed from the noisy signal. The L -cluster- M -best, the neighborhood Gaussian selection and the subspace clustering method are evaluated. The percentage of Gaussians calculated of the first method is controlled by the M to L ratio, while that of the second is controlled by the average ratio of the neighborhood size to the number mixtures, which is in turn controlled by a threshold θ , a parameter which is initialized to 1 and is increased to achieve the predefined average neighborhood size. It is observed in Aurora-4 that the performance does not get much better as the number of clusters L is increased. We found $L = 110$ to be a reasonable value. In our experiments, replacing the score of the unselected member Gaussians with a very small value gives better results than assigning their scores with those of their cluster Gaussians in both L -cluster- M -best and neighborhood implementation which confirms the findings of [8].

In Figure 7), the average fraction of Gaussians to be calculated gives an indication of the computational efficiency of decoding. Curve (a) shows the tradeoff between WER using oracle masks and the percentage of Gaussians calculated of 110-cluster- M -best Gaussian selection method, where $M = 11, 22, \dots, 99, 110$ yielding the ten data points along the curve. Curve (b) shows that of the neighborhood method. The average ratio of neighborhood size to the number of mixtures equals 10%, 20% ... 90%. The L -cluster- M -best seems the more efficient method. This can be explained as follows: the L -cluster- M -best method selects several clusters around the observation, while the neighborhood method selects Gaussians in a wide area, but not centred around the observation, making the former the more effective method. Curves (a) and (b) reach the same point when 100% of the Gaussians are calculated. It also shows that this point exhibits a higher error rate than most points along curve (a), i.e. the Gaussian selection also improves accuracy, a phenomenon also observed in [38]. Curve (c) is the performance of subspace clustering with 4096, 6144 and 8192 clusters in each of the static and dynamic streams. We observe that this method does not

perform as well as the cluster-based methods. The MDT-constraint to use full spectral feature streams as subspaces for the sake of the accuracy of the imputed values leads to suboptimal clustering, for optimal subspaces are typically of a lower dimension [38]. Using subspaces of low dimension would mean that the correlations among features are disregarded in the MDT imputation, leading to severe loss in accuracy. In order to achieve an efficient computation without losing much accuracy (or even gaining accuracy), we choose L -cluster- M -best as the Gaussian selection method in further experiments.

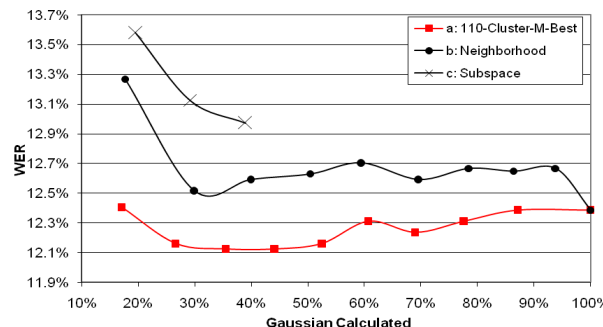


Figure 7: WER with percentage of Gaussian calculated of L -cluster- M -best, neighbourhood Gaussian selection and subspace clustering in Aurora-4 experiments

5.4.2. Experiments on the SpeechDat-Car Flemish Database

Figure 9) shows the accuracy results on the recognition task with 637 active words or short commands. The small gain in accuracy due to the 170-cluster-34-best Gaussian selection method is also observed here. The reason for using 170 clusters here is the larger number of Gaussians compared to the Aurora-4 task.

Figure 8) compares the CPU time per frame of the 170-cluster-34-best Gaussian selection with the baseline recognizer without removing Gaussians on the isolated words task. Gaussian selection saves about 60% CPU time. Both the CPU time of the baseline system and Gaussian selection are increased as the SNR is decreased, which is attributed to the increase in number of unreliable time-frequency cells, making the NNLSQ more expensive to solve. Furthermore, as more time-frequency cells are unreliable and the noisy observations are constraining the NNLSQ problem less ($s \leq y$) at lower SNR, the larger clusters tend to end up more in the M -best list, resulting in more Gaussians to be evaluated. Finally, noise addition reduces the likelihood contrasts in the search space, which makes its pruning mechanisms less effective.

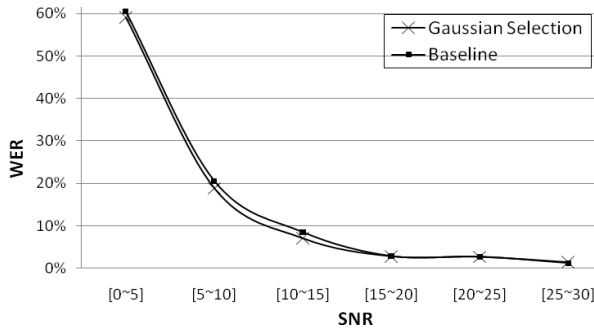


Figure 9: WER in % as a function of SNR (in 5dB bins) of 170-Cluster-34-Best Gaussian selection and the baseline recogniser without Gaussian removal on isolated words grammar of MIDAS Flemish in-car data

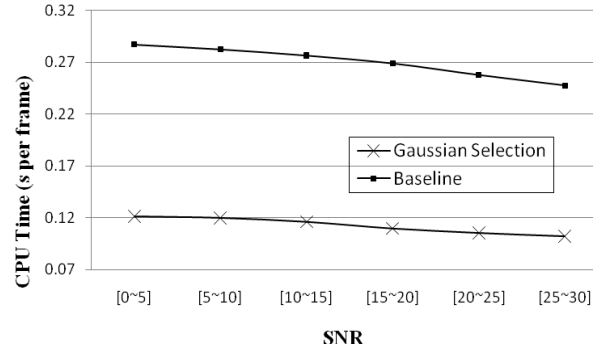


Figure 8: CPU time per frame of 170-Cluster-34-Best Gaussian selection and the baseline recogniser without Gaussian removal on isolated word grammar of the Flemish SpeechDat in-car data.

6. Conclusions

When applying MDT to ASR, it is not necessary to restrict the acoustic models to a mixture of Gaussians with diagonal covariance in the (log)-spectral domain. Linear transformations can be handled as well, though the complexity of the resulting non-negative least squares problem is increased considerably. With a careful design of this linear transform, PROSPECT features have been defined which maintain the accuracy but result in a first gain of about a factor of about 2 compared to a MDT-system based on cepstral features. Further gains in speed of a factor of 3 can be achieved using a L -cluster- M -best Gaussian selection method, which turned out to be the most effective approach.

A critical issue for robust and accurate MDT-based recognition is the estimation of the reliability mask. We have presented a method which models typical speech spectra using a vector quantization description and which relies on harmonicity of the speech signal. We have compared the obtained accuracy and found that the additional constraints imposed by the model result in substantial accuracy gains.

Despite the gains in speed obtained with the presented techniques, applying MDT to large vocabulary tasks remains computationally intensive.

7. Acknowledgements

This research is supported by the MIDAS project of the Nederlandse Taalunie under the STEVIN program.

References

- [1] M. Cooke M, Ph. Green, L. Josifovski, A. Vizinho, "Robust automatic speech recognition with missing and unreliable acoustic data", *Speech Communication* 34 (2001), pp. 267-285.
- [2] L. Josifovski, "Robust automatic speech recognition with missing and unreliable data", PhD. Dissertation, University of Sheffield, 2002..
- [3] F. Faubel, J. McDonough, and D. Klakow, "Bounded conditional mean imputation with Gaussian mixture models: A reconstruction approach to partly occluded features," in *Proc. International Conference on Acoustics, Speech and Signal Processing*, Taipei, Taiwan, Apr. 2009, pp. 3869-3872.
- [4] A. Gersho and R.M. Gray. Vector quantization and signal compression. Kluwer Academic Press, 1992.
- [5] S. So and K. K. Paliwal. "Improved noise-robustness in distributed speech recognition via perceptually-weighted vector quantisation of filterbank energies", In *Proc. International Conference on Spoken Language Processing*, pp. 941-944, Lisbon, Portugal, September 2005.
- [6] D. P. W. Ellis and R. J. Weiss. "Model-based monaural source separation using a vector-quantized phase-vocoder representation", in *Proc. International Conference on Acoustics, Speech and Signal Processing*, Toulouse, France, 2006.
- [7] H. Van hamme. "Robust speech recognition using cepstral domain missing data techniques and noisy masks", In *Proc. International Conference on Acoustics, Speech and Signal Processing*, Montreal, Canada, May 2004, pp. 213-216
- [8] M. L. Seltzer, B. Raj, and R.M. Stern. "Classifier-based mask estimation for missing feature methods of robust speech recognition", in *Proc. International Conference on Spoken Language Processing*, Beijing, China, October 2000, pp. 538-541
- [9] M. Van Segbroeck and H. Van hamme. "Vector-Quantization based mask estimation for missing data automatic speech recognition", in *Proc. International Conference on Spoken Language Processing*, Antwerp, Belgium, August 2007, pp. 910-913.
- [10] D. J. Hermes, "Measurement of pitch by subharmonic summation", *J. Acoust. Soc Am.*, 83 (1), pp. 257-264, Jan. 1988
- [11] C. T. Kelley, *Iterative Methods for Optimization*, SIAM Frontiers in Applied Mathematics, no 18, 1999
- [12] M. Wu and D. Wang and G. J. Brown, "A Multi-Pitch Tracking Algorithm for Noisy Speech", *IEEE Transactions on Speech and Audio Processing*, vol 11, pp. 229-241, 2002
- [13] D. Ealey, H. Kelleher, D. Pearce, "Harmonic tunneling: tracking non-stationary noises during speech", in *Proc. Eurospeech*, Aalborg, Denmark, pp. 437-410, Sep. 1999
- [14] R. Martin. "Noise power spectral density estimation based on optimal smoothing and minimum statistics", In *IEEE Transactions on Speech and Audio Processing*, volume 9, pp. 504-512, July 2001.
- [15] J. Ramirez, J.M. Górriz, J.C. Segura, C.G. Puntonet, and A. Rubio. "Speech/non-speech discrimination based on contextual information integrated bispectrum LRT", In *IEEE Signal Processing Letters*, 2006.
- [16] S. Demange, C. Cerisara, and J.-P. Haton. "Missing data mask estimation with frequency and temporal dependencies", *Computer, Speech and Language*, 23 (1):25-41, July 2009.
- [17] P. Renevey and A. Drygajlo, "Missing feature theory and probabilistic estimation of clean speech components for robust speech recognition," in *Proc. Eurospeech*, Budapest, Hungary, 1999, pp. 2627-2630
- [18] J. Barker, L. Josifovski, M. Cooke, and Ph. Green, "Soft decisions in missing data techniques for robust speech recognition", in *Proc. International Conference on Spoken Language Processing*, Beijing, China, Oct. 2000, pp. 373-376.

- [19] S. Yamamoto, J.-M. Valin, K. Nakadai, J. Rouat, F. Michaud, T. Ogata, H. Okuno, "Enhanced robot speech recognition based on microphone array source separation and missing feature theory." In *Proc. Interspeech*, Lisbon, Portugal, September 2005
- [20] M. Van Segbroeck and H. Van hamme, "Robust speech recognition using missing data techniques in the prospect domain and fuzzy masks", in *Proc. International Conference on Acoustics, Speech and Signal Processing*, Las Vegas, U.S.A., Apr. 2008, pp. 4393-4396.
- [21] K. Demuynck. "Extracting, Modelling and Combining Information in Speech Recognition". PhD thesis, K.U.Leuven, ESAT, February 2001
- [22] M. Van Segbroeck and H. Van hamme, "Handling Convolutional Noise in Missing Data Automatic Speech Recognition", in *Proc. International Conference on Spoken Language Processing*, pp. 2562-2565, Pittsburgh, U.S.A., September 2006.
- [23] N. Oostdijk, "The Spoken Dutch Corpus. Overview and first evaluation", in *Proc. International Conference on Language Resources and Evaluation*, pp. 887-894, 2000
- [24] H. Van hamme, "Handling time-derivative features in a missing data framework for robust automatic speech recognition," in *Proc. International Conference on Acoustics, Speech and Signal Processing*, Toulouse, France, May 2006, pp. 293-296.
- [25] C. Cerisara, S. Demange and J.-P. Haton. "On noise masking for automatic missing data recognition: a survey and discussion". *Computer, Speech and Language*, 21(3), pp. 443-457, July 2007
- [26] J. Barker, M. Cooke and P. Green, "Robust ASR based on clean speech models: an evaluation of missing data techniques for connected digit recognition in noise", in *Proc. Eurospeech*, Aalborg, Denmark, Sep. 2001.
- [27] M. Seltzer, B. Raj and R. M. Stern, "A Bayesian classifier for spectrographic mask estimation for missing feature speech recognition", *Speech Communication* 43 (2004), pp. 379-393
- [28] H. Van hamme, "PROSPECT Features and their Application to Missing Data Techniques for Robust Speech Recognition", in *Proc. International Conference on Speech and Language Processing*, volume I, 101-104, 2004
- [29] I. Rogina and A. Waibel, "Learning state-dependent stream weights for multi-codebook HMM speech recognition systems", in *Proc. International Conference on Acoustics, Speech and Signal Processing*, Adelaide, Australia, Apr. 1994, pp. 217-220.
- [30] F. Jurgen, R. Ivica, "The bucket box intersection (BBI) algorithm for fast approximative evaluation of diagonal mixture Gaussians", in *Proc. International Conference on Acoustics Speech and Signal Processing*, 1996, pp. 837-840
- [31] K. Demuynck, J. Duchateau and D. Van Compernelle. "Reduced Semi-continuous Models for Large Vocabulary Continuous Speech Recognition in Dutch", in *Proc. International Conference on Spoken Language Processing*, Philadelphia, U.S.A., October 1996, volume IV, pp. 2289-2292.
- [32] T. A. Myrvoll and F. K. Soong, "Optimal Clustering of Multivariate Normal Distributions Using Divergence and Its Application to HMM Adaptation", in *Proc. International Conference on Acoustics Speech and Signal Processing*, 2003, volume I, pp. 552-555.
- [33] T. Watanabe, K. Shinoda, K. Takagi, K.-I. Iso, "High Speed Speech Recognition Using Tree-Structured Probability Density Function", in *Proc. International Conference on Acoustics Speech and Signal Processing*, 1995, Volume I, pp. 556 - 559.
- [34] Bocchieri, E., "Vector quantization for efficient computation of continuous density likelihoods", *Proc. ICASSP*, Volume 2, 692-695, 1993.
- [35] J. Olsen, "Gaussian selection using multiple quantization indexes," *IEEE Nordic Processing Symposium*, 2000
- [36] N. Parihar and J. Picone., "An Analysis of the Aurora Large Vocabulary Evaluation," in *Proc. Eurospeech*, 2003, pp. 337-340.
- [37] A. Moreno, B. Lindberg, C. Draxler, G. Richard, K. Choukri, J. Allen ND S. Euler, "Speechdat-car: A large speech database for automotive environments", in *Proc. LREC*, 2000.
- [38] E. Bocchieri, B.K.-W. Mak, "Subspace Distribution Clustering Hidden Markov Model", *IEEE Trans. Speech and Audio Proc.*, 9(3):264-275, 2001.