EURASIP Journal on Audio, Speech, and Music Processing
a SpringerOpen Journal

RESEARCH Open Access

Multi-candidate missing data imputation for robust speech recognition

Yujun Wang* and Hugo Van hamme

Abstract

The application of Missing Data Techniques (MDT) to increase the noise robustness of HMM/GMM-based large vocabulary speech recognizers is hampered by a large computational burden. The likelihood evaluations imply solving many constrained least squares (CLSQ) optimization problems. As an alternative, researchers have proposed frontend MDT or have made oversimplifying independence assumptions for the backend acoustic model. In this article, we propose a fast Multi-Candidate (MC) approach that solves the per-Gaussian CLSQ problems approximately by selecting the best from a small set of candidate solutions, which are generated as the MDT solutions on a reduced set of cluster Gaussians. Experiments show that the MC MDT runs equally fast as the uncompensated recognizer while achieving the accuracy of the full backend optimization approach. The experiments also show that exploiting the more accurate acoustic model of the backend does pay off in terms of accuracy when compared to frontend MDT.

Keywords: speech recognition, constrained optimization, missing data, noise robustness

1. Introduction

One of the major concerns in deploying Automatic Speech Recognition (ASR) applications is the lack of robustness of the technology when compared to human listeners. A key aspect is the sensitivity to background noise. This effect is caused by the differences between the conditions in which the statistical models for speech are trained and those in which they are applied in real-life situations. Many approaches which reduce the mismatch to improve the noise robustness of speech recognition have been proposed earlier. They modify either the frontend signal preprocessing or the backend acoustic model of the recognizer. A popular frontend method is the Advanced Front-End [1] which applies multiple stages of Wiener filtering to remove the background noise from the corrupted observations. Other techniques working in the frontend are, e.g., spectral subtraction [2], Stereo Piecewise Linear Compensation for Environment [3] and the Vector Taylor series compensation algorithm [4]. Some examples of backend approaches are Parallel Model Combination (PMC) [5] and model adaption algorithms, such as Maximum Likelihood Linear Regression (MLLR) [6] and Maximum A Posterior probability (MAP) based adaptation [7].

In the late 1990s, Missing Data Techniques (MDT) were introduced in speech recognition as a perceptually motivated approach to improve the noise robustness of a speech recognizer. Research in Auditory Scene Analysis (ASA) [8] proposed models for the capability of human listeners to deal with concurrent signals. The human auditory system is able to extract sufficient information from the speech source of interest in order to recognize what is said, even if parts of the target signal are masked by other signals. It exploits the redundancy in the speech signal and can thus handle missing data. The motivation of MDT is to explore these capabilities of human listeners and exploit them in ASR to reduce the performance gap between humans and computers. It relies on the model that a given spectral band at a given time is dominated by either speech or noise. In the frontend preprocessing, the time-frequency regions of a speech signal are labeled as reliable or as unreliable. This labeling information is encoded into a so-called *missing data mask*. In the backend decoding, features in the unreliable regions are either ignored or predicted to alleviate the mismatch. This compensation strategy relies only on the speech model and unlike PMC for instance, it does not require a model of the noise, though

* Correspondence: Yujun.Wang@esat.kuleuven.be
Department of ESAT, Katholieke Universiteit Leuven, Kasteelpark Arenberg 10, B-3001 Leuven, Belgium

Springer

© 2012 Wang and Van hamme; licensee Springer. This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/2.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

some assumptions about the noise are required instead while generating the missing data mask [9,10]. In recent years, the MDT was extended to techniques such as the glimpsing model [11] and speech fragment decoding [12]. Other related work includes the propagation of uncertainty [13] where the authors transform the uncertainty encoded in the binary mask from the spectral domain to the cepstral domain, and handle the transformed uncertainty with the cepstral backend acoustic models. The authors of [14] introduce a two-pass MDT system, where the lattice generated by the MDT recognizer in the first pass is rescored. In the second pass, a state-based hypothesis test then generates the so-called "integrated mask", yielding better recognition results.

The two major problems in MDT are first estimating the mask and then exploiting these masks during recognition. Identifying the 'missing' part during recognition is an essential step in MDT as proposed by Cooke et al. [15] and Lippmann and Carlson [16]. A missing data detector makes a binary decision about which spectro-temporal components are unreliable due to noise distortions and which remain reliable, i.e., are dominated by speech. Approaches of missing data mask estimation, such as Bayesian classification [17], harmonic mask estimation [10], local SNR-based mask estimation [18,19], and VQ mask estimation [9] mainly exploit characteristics of the speech signal. The authors of [20] estimate the missing data masks based on computational ASA. More approaches can be found in a survey on missing data mask estimation by Cerisara et al. [21]. The concept of binary reliability masks can be extended to *soft masks* [22] when uncertainty about the reliability information is taken into account. The mask then assumes continuous values instead of binary values. Soft masks are not considered in this article, as we have found them to provide little benefit in [23].

Several paradigms have been designed to apply MDT once the masks are computed. MDT was first formulated for a spectral acoustic model [15], which is referred to as *spectral MDT* in this article. The spectral energy within each unreliable component can be either reconstructed based on the acoustic model and the reliable information, or marginalized out of the probability density functions (PDF) of the HMM states. The former scheme is defined as *imputation* and the latter is defined as *marginalization*. In order to improve the performance of MDT, Raj et al. [24], Van hamme [25], Cerisara [26], Häkkinen and Haverinen [27], and Faubel et al. [28] applied MDT using cepstral acoustic models, which are referred to as *cepstral MDT* in this article. The experimental results of cepstral MDT demonstrate its advantage over the spectral model. The authors of [24] used MDT imputation to enhance the speech features in the front-end, while in Maximum Likelihood (ML) Gaussian-based imputation [25] and in conditional mean imputation [28], the authors consider MDT imputation associated with Gaussians in the backend.

The above work addresses the robustness of the MDT system rather than its efficiency. MDT systems involve much more intensive computation in the backend, as explained in Section 3. This was already noticed in [15], where the problem was addressed by compromising on the acoustic model (diagonal Gaussians for log-spectral features). An alternative solution is to formulate MDT as a front-end technique [24]. In this article, we propose a Multi-Candidate (MC) MDT which not only produces competitive recognition accuracy, but also possesses the same efficiency as a conventional large vocabulary recognizer under noisy conditions. We advocate the backend approach, since it exploits the most accurate speech model that is available in the recognizer to compensate for the missing data. Each HMM state represents an accurate hypothesis about what the missing speech could be, integrating all knowledge that is available in the decoder: acoustics, lexical information, and language model. Hence, we expect more accurate missing data imputation than with frontend MDT approaches, where such sophistication is not available. In our setting, we go beyond the state level and compute a clean speech vector per Gaussian. In addition to the entire set of Gaussians embedded in the HMM, a fairly small set of Gaussians are trained to function as cluster Gaussians (CG). They provide feasible candidates (i.e., they satisfy the constraints for the imputed data, as described in Section 2.1) of imputations for the entire set of Gaussians. As such, instead of solving the full optimization problem for each Gaussian in the acoustic model, candidate solutions are selected from the CG and the most likely one is retained. Therefore, implementation of MC MDT requires only a modest modification of conventional HMM-based recognizers. The MC MDT forms the main contribution of this article. It is an algorithm that aims at *computational gains for large vocabulary speech recognizers without sacrificing accuracy or robustness*. It provides a solution for applying MDT to an existing backend model trained for the speech feature vector of one's choice. Furthermore, we show experimentally that we gain more immunity to noise than if MDT is applied as a frontend feature-enhancement technique [24] and compare several methods for solving the imputation problem.

Figure 1 shows the architecture of the MC MDT system. This article is focused on the three blocks in the middle, i.e., the imputation of the (cluster of) CG and MC imputation for the backend Gaussians (BGs). The rest of this article is arranged as follows: in Section 2, we introduce the conventional state-based imputation and marginalization [15] as well as the spectral reconstruction [24]. In Section 3, we discuss MDT imputation under the

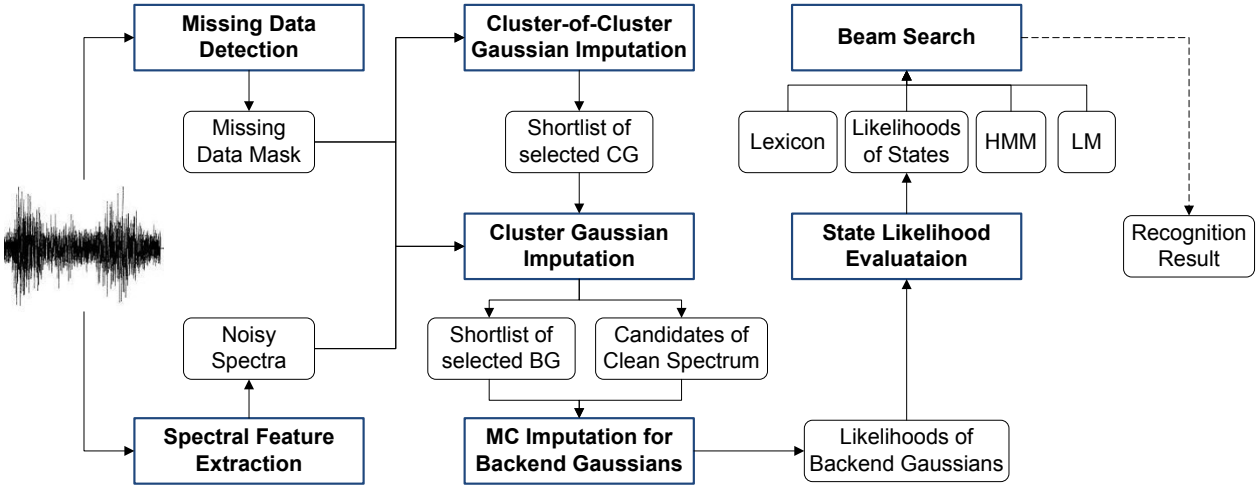

Figure 1 The block diagram of the MC MDT system. The noisy spectrum is extracted and the missing data mask is estimated from the speech waveform. Two layers of CG, namely the CG and the CCG are trained before decoding. For each frame, the imputation is performed firstly on the CCGs using the mask and the noisy observation. Their likelihoods are calculated on the corresponding imputed spectrum. A shortlist of selected CGs is derived from these likelihoods. The imputed clean spectrum of each selected CG is then calculated based again on the noisy observation and the mask. The likelihoods of the CGs are calculated on their imputed spectra and used to determine which BGs are worth calculating. Each selected BG is evaluated on multiple related imputed spectra proposed by the CGs and the largest likelihood is retained as the final BG likelihood. The beam search can then proceed as in a conventional HMM system, integrating acoustical evidence, lexical information and a language model to generate the recognition result.

framework of ML decoding and why it becomes difficult when using a model trained with decorrelated features such as cepstral features or features generated by, e.g., linear discriminant analysis (LDA) [29]. Section 4 describes the approach of MC MDT imputation using CG to speed up the Gaussian-based imputation. Section 5 explains how to further speed up the imputation of the CG by selecting a subset of the CG dynamically. Section 6 describes several experimental results. Finally, in Section 7 we present our conclusions and propose future work.

2. Spectral and Cepstral MDT systems

In this section, we review some of the concepts of MDT that lead to approaches that are most related to the proposed system.

2.1. Bounds

Environmental noises are assumed to be additive in the spectral domain. Hence, at frame t, the log-spectra of the underlying complete clean speech $\mathbf{x}_t$ can be assumed to be approximately bounded above by the observed noisy feature vector $\mathbf{y}_t$, namely:

$$\mathbf{x}_t \leq \mathbf{y}_t \tag{1}$$

where the inequality sign for vectors applies component-wise. Both $\mathbf{x}_t$ and $\mathbf{y}_t$ can be partitioned into their reliable and unreliable sub-vectors according to the mask:

$$\mathbf{x}_t = \begin{bmatrix} \mathbf{x}_{t,r} \\ \mathbf{x}_{t,u} \end{bmatrix} \text{ and } \mathbf{y}_t = \begin{bmatrix} \mathbf{y}_{t,r} \\ \mathbf{y}_{t,u} \end{bmatrix}$$

For the reliable spectro-temporal regions, the observed noisy features are deemed to be pure speech:

$$\mathbf{x}_{t,r} = \mathbf{y}_{t,r},$$

whereas for the unreliable regions, the observed features act merely as upper bounds for the clean speech:

$$\mathbf{x}_{t,u} \leq \mathbf{y}_{t,u}$$

2.2. State-based imputation and marginalization

The authors of [15] formulated several MDT approaches which use the acoustic models trained in the same domain in which the masks are expressed and in which the constraints of Equation (1) hold. In their experiments, the acoustic feature vectors are obtained via a 64-channel auditory filter bank with center frequencies spaced linearly on an ERB scale from 50 Hz to 8 kHz. The HMM-based speech recognizer is adapted to accommodate MDT by modifying the state likelihood evaluation as outlined below. Each HMM state is expressed as a mixture of multivariate Gaussians with a diagonal covariance matrix. The MDT here is carried out frame-by-frame and is assumed independent across frames. The authors proposed both state-based imputation and marginalization. Besides the upper bound $\mathbf{y}_{t,\ u}$, a lower bound can also be applied to

control the arbitrariness of compensation for the unreliable components. This idea can be applied to all methods described in this article. However, for consistency, we will omit lower bounds from this article.

In state-based marginalization, each state output PDF is a function of the reliable components only, while the unreliable components are marginalized out, i.e., each unreliable component is integrated over the range of values it can assume. The PDF of a state s is given by:

$$p(\mathbf{y}_{t,r}|s) = P(k|s) \sum_{k \in \mathrm{G}\ (s)} p(\mathbf{y}_{t,r}|k,s) \int_{-\infty}^{\mathbf{y}_{t,u}} p(\mathbf{x}_{t,u}|k,s) d\mathbf{x}_{t,u}$$

where $G(s)$ represents the set of Gaussians belonging to the Gaussian Mixture Model (GMM) of state s. The integral of Gaussian k can be calculated using the component-wise error function because its covariance matrix is assumed to be diagonal.

In state-based imputation, the clean speech is imputed for every state s, followed by calculating the likelihoods using the imputed values, which will be utilized to expand hypotheses in the search space during decoding. Two ways of imputing the clean speech per state are given: linear combination or winner-takes-all.

In linear combination, the Minimum Mean Squar Error (MMSE) estimate of the imputation from state s is

$$\hat{\mathbf{x}}_{t,u,s} = \sum_{k \in G(s)} P(k|\mathbf{y}_{t,r}, s) \boldsymbol{\mu}_{u,k}$$

where $\boldsymbol{\mu}_{u,\ k}$ is the unreliable sub-vector of mean of Gaussian k and

$$P(k|\mathbf{y}_{t,r}, s) = \frac{P(k|s) p(\mathbf{y}_{t,r}|k) \int_{-\infty}^{\mathbf{y}_{t,u}} p(\mathbf{x}_{t,u}|k) d\mathbf{x}_{t,u}}{\sum_{j \in G(s)} P(j|s) p(\mathbf{y}_{t,r}|j) \int_{-\infty}^{\mathbf{y}_{t,u}} p(\mathbf{x}_{t,u}|j) d\mathbf{x}_{t,u}}$$

In winner-takes-all, after the clean speech is imputed for each Gaussian belonging to state s, the mixture's likelihood is evaluated for all imputed values and the most likely imputation is selected as the imputation of the state. In other words, the imputation of state s is approximated by the clean speech vector imputed from its $\hat{k}$ th member Gaussian:

$$\hat{\mathbf{x}}_{t,u,s} \cong \hat{\mathbf{x}}_{t,u,\hat{k}}$$

where $\hat{k} = \arg\max_{k \in G(s)} p(\hat{\mathbf{x}}_{t,u,k}|s)$. $\hat{\mathbf{x}}_{t,u,k}$ is the maximum likelihood imputation of the unreliable subsector $\mathbf{x}_{t,\ u,\ k}$ for Gaussian k included in G(s):

$$\hat{\mathbf{x}}_{t,u,k} = \arg\max_{\mathbf{x}_{t,u,k} \le \mathbf{y}_{t,u}} p(\mathbf{x}_{t,u,k}|k) \quad k \in \mathrm{G}\ (s)$$

This problem has a closed form solution:

$$\hat{\mathbf{x}}_{t,u,k} = \begin{cases} \boldsymbol{\mu}_{u,k} \text{ if } \boldsymbol{\mu}_{u,k} \le \mathbf{y}_{t,u} \\ \mathbf{y}_{t,u} \text{ if } \boldsymbol{\mu}_{u,k} > \mathbf{y}_{t,u} \end{cases} \quad k \in \mathrm{G}\ (s) \tag{2}$$

where it should be understood that we have written the solution vectorially for convenience, but the top or bottom case in (2) may apply to different vector components. The imputation using a spectral acoustic model containing Gaussians with a diagonal covariance matrix has an analytical solution because the components of the log-spectral features are considered to be independent. However, the spectral features do have correlation among their components and the spectral GMM used above is not very effective to model this. The performance of HMM speech recognizers using GMMs with diagonal covariance is significantly better when using decorrelated features, e.g., MEL Frequency Cepstral Coefficients (MFCC). Therefore, a cepstral MDT model with diagonal covariance-Gaussians is introduced in the following section.

2.3. Spectral reconstruction

In [24], the authors reconstruct the spectral features using either a correlation-based method or a cluster-based method. The reconstructed spectra are then transformed into cepstra for processing by the speech recognizer.

The correlation-based approach solves the imputation of unreliable components at each frame by exploiting the correlations among the components in the spectro-temporal representation. The correlation is modeled by a Gaussian wide-sense stationary (WSS) process whose parameters are learned from training data. The core of the algorithm is a bounded MAP estimate:

$$\hat{\mathbf{x}}_{t,u} = \arg\max_{\mathbf{x}_{t,u} \le \mathbf{y}_{t,u}} p(\mathbf{x}_{t,u}|\mathbf{y}_{t,n}) \tag{3}$$

where $\mathbf{y}_{t,\ n}$ is the neighborhood vector containing all the related reliable components which are spectrally and temporally sufficiently close to $\mathbf{x}_{t,\ u}$ as defined by the WSS model. The likelihood $p(\mathbf{x}_{t,\ u}|\mathbf{y}_{t,\ n})$ is modeled with a full covariance Gaussian conditioned on the observed $\mathbf{y}_{t,\ n}$. The authors establish an iterative approach to solve (3).

In the cluster-based approach, the distribution of the observation is modeled by a spectral GMM with M mixture components with full covariance. Each of these mixture components is called *CG*, trained by the Expectation-Maximization (EM) algorithm. The unreliable components of the reconstructed spectra are obtained from a linear combination of the values imputed for the CG:

$$\hat{\mathbf{x}}_{t,u} = \sum_{m=1}^{M} P(m \,|\mathbf{x}_{t,u,m} \le \mathbf{y}_{t,u}, \mathbf{y}_{t,r})\hat{\mathbf{x}}_{t,u,m} \tag{4}$$

where

$$\hat{\mathbf{x}}_{t,u,m} = \underset{\mathbf{x}_{t,u,m} \le \mathbf{y}_{t,u}}{\arg\max}\, p(\mathbf{x}_{t,u,m}|\mathbf{y}_{t,r}, m) \tag{5}$$

is the imputation resulting from the mth CG, a bounded optimization problem which can be solved by the MAP algorithm as in the correlation-based approach. $P(m|\mathbf{x}_{t,\,u,\,m} \le \mathbf{y}_{t,\,u}, \mathbf{y}_{t,\,r})$ is the posterior probability of the CG given the reliable data and the feasible region for the unreliable data.

$$P(m \,|\mathbf{x}_{t,u,m} \le \mathbf{y}_{t,u}, \mathbf{y}_{t,r}) = \frac{P(m)\int_{-\infty}^{\mathbf{y}_{t,u}} P(\mathbf{x}_{t,u,m}, \mathbf{y}_{t,r}|m)d\mathbf{x}_{t,u,m}}{\sum_{j=1}^{M} P(j)\int_{-\infty}^{\mathbf{y}_{t,u}} p(\mathbf{x}_{t,u,j}, \mathbf{y}_{t,r}\,|j)d\mathbf{x}_{t,u,j}} \tag{6}$$

To make computation of this posterior probability tractable, the spectral CGs are assumed to be diagonal in this circumstance.

In both the correlation-based and the cluster-based method, the reconstruction is separated from the decoding and there is only one single imputation per frame, while in the spectral state-based imputation of Section 2.2, each state or Gaussian has its own imputation, which is theoretically more suitable for an ML-based recognizer. The likelihood of each state is calculated at its imputed value and used in the backend of the recognizer which incorporates the lexical and grammatical knowledge to drive the path pruning in the beam.

It should be noted that the authors of [15] show that state-based marginalization outperforms state-based imputation. Therefore, it would be natural to formulate marginalization for cepstral or other decorrelated models as well. However, this leads to definite integration of full covariance Gaussians. Even if approximations described in [28] would be applied to marginalization, the computational complexity is not acceptable for a practical speech recognizer. Hence, we only focus on imputation with decorrelated models.

3. Missing data imputation for maximum likelihood decoding

State-of-the-art automatic speech recognizers take a Bayesian approach, i.e., the decoding process is to find a sequence of words $\hat{\mathbf{W}}$ whose posterior probability is maximal given a T-frame sequence of observations $\mathbf{y}_{1\ldots T}$:

$$\hat{\mathbf{W}} = \underset{\mathbf{W}}{\arg\max}\, P(\mathbf{W}|\mathbf{y}_{1\cdots T}) \approx \underset{\mathbf{W}}{\arg\max}\, p(\mathbf{y}_{1\cdots T}|s_{1\cdots T})P(\mathbf{W})$$

where the language model $P(\mathbf{W})$ is the probability of a hypothesized word sequence $\mathbf{W}$. In practice, the most likely state sequence $s_{1\ldots T}$ that realizes $\mathbf{W}$ is found. In MDT, the maximization should be additionally taken over the unreliable features to be imputed, i.e., $\mathbf{x}_{1\ldots T,\,u}$, to find out the optimal imputation $\hat{\mathbf{x}}_{1\ldots T,u}$ bounded by the noisy observation $\mathbf{y}_{1\ldots T,\,u}$.

$$(\hat{\mathbf{W}}, \hat{\mathbf{x}}_{1\ldots T,u}) = \underset{\mathbf{W}, \mathbf{x}_{1\ldots T,u} \le \mathbf{y}_{1\ldots T,u}}{\arg\max}\; p(\mathbf{x}_{1\cdots T,u}, \mathbf{y}_{1\ldots T,r}|s_{1\cdots T})P(\mathbf{W}).$$

For a given state sequence $s_{1\ldots T}$ with $\mathbf{W}$ embedded, the complete speech is given by the following expression, where we have assumed state-conditional independence of $\mathbf{x}_{1\ldots T,\,u}$:

$$\hat{\mathbf{x}}_{s_{1\ldots T,u}} = \underset{\mathbf{x}_{1\ldots T,u} \le \mathbf{y}_{1\ldots T,u}}{\arg\max}\; p(\mathbf{x}_{1\cdots T,u}, \mathbf{y}_{1\ldots T,r}|s_{1\cdots T}) = a\prod_{t=1}^{T} \underset{\mathbf{x}_{t,u} \le \mathbf{y}_{t,u}}{\arg\max}\, p(\mathbf{x}_{t,u}, \mathbf{y}_{t,r}|s_t) \tag{7}$$

where a is the product of the transition probabilities between the states on the hypothesized path. The maximization in Equation (7) can be accomplished frame-by-frame, i.e., the optimal clean speech at time t is obtained by the maximization of the output PDF of state s over the complete speech $\mathbf{x}_t$ bounded by the observation $\mathbf{y}_t$:

$$\hat{\mathbf{x}}_{t,u,s} = \underset{\mathbf{x}_{t,u,s} \le \mathbf{y}_{t,u}}{\arg\max}\, p(\mathbf{x}_{t,u,s}, \mathbf{y}_{t,r}|s) = \underset{\mathbf{x}_{t,u,s} \le \mathbf{y}_{t,u}}{\arg\max} \sum_{k \in G(s)} P(k|s)p(\mathbf{x}_{t,u,s}, \mathbf{y}_{t,r}|k) \tag{8}$$

Equation (8) formulates an ML state-based missing data imputation. The constrained optimization in (8) is not computationally tractable. If each member Gaussian in a state output PDF is assumed to impute its own clean speech using MLE:

$$\hat{\mathbf{x}}_{t,u,k} = \underset{\mathbf{x}_{t,u,k} \le \mathbf{y}_{t,u}}{\arg\max}\, p(\mathbf{x}_{t,u,k}, \mathbf{y}_{t,r}|k) \tag{9}$$

MDT imputation becomes ML Gaussian-based imputation, which is an approximation of the state-based imputation but is computationally more tractable. It will be shown in Section 6.4.4 that (8) and (9) yield comparable recognition accuracy.

If the model used for imputation is trained with cepstral features or other decorrelated features, such as LDA [29] or HLDA [30] features, Gaussian k can be formulated in the log-spectral domain after the corresponding linear transformation $\mathbf{C}$ of full row-rank is applied:

$$\begin{aligned} p(\mathbf{x}_{t,k}|k) &= \frac{1}{\sqrt{(2\pi)^{D_m}|\boldsymbol{\Sigma}_k|}} \exp\left(-\frac{1}{2}(\mathbf{C}\mathbf{x}_{t,k} - \boldsymbol{\mu}_k)'\boldsymbol{\Sigma}_k^{-1}(\mathbf{C}\mathbf{x}_{t,k} - \boldsymbol{\mu}_k)\right) \\ &= \frac{1}{\sqrt{(2\pi)^{D_m}|\boldsymbol{\Sigma}_k|}} \exp\left(-\frac{1}{2}(\mathbf{x}_{t,k} - \mathbf{C}^{+}\boldsymbol{\mu}_k)'\mathbf{C}'\boldsymbol{\Sigma}_k^{-1}\mathbf{C}(\mathbf{x}_{t,k} - \mathbf{C}^{+}\boldsymbol{\mu}_k)\right) \end{aligned}$$

where $\mathbf{C}^{+}$ represents the pseudo inverse of $\mathbf{C}$, $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$ are the mean and diagonal covariance of the transformed features and D_m denotes the dimension of the decorrelated feature vectors. Instead of maximizing probabilities, we can equivalently minimize the cost function:

$$Q_k = (\mathbf{x}_{t,k} - \mathbf{C}^+\boldsymbol{\mu}_k)'\mathbf{C}'\boldsymbol{\Sigma}_k{}^{-1}\mathbf{C}(\mathbf{x}_{t,k} - \mathbf{C}^+\boldsymbol{\mu}_k) \quad (10)$$

with the precision matrix

$$\mathbf{H}_k = \mathbf{C}'\boldsymbol{\Sigma}_k{}^{-1}\mathbf{C}$$

the maximization of (9) over $\mathbf{x}_{t,\,k}$ becomes:

$$\hat{\mathbf{x}}_{t,u,k} = \underset{\mathbf{x}_{t,u,k} \le \mathbf{y}_{t,u}}{\arg\min} \left(\begin{bmatrix} \mathbf{x}_{t,u,k} \\ \mathbf{y}_{t,r} \end{bmatrix} - \mathbf{C}^+\boldsymbol{\mu}_k \right)' \mathbf{H}_k \left(\begin{bmatrix} \mathbf{x}_{t,u,k} \\ \mathbf{y}_{t,r} \end{bmatrix} - \mathbf{C}^+\boldsymbol{\mu}_k \right) \quad (11)$$

Notice that $\mathbf{H}_k$ can be singular (e.g., when the cepstral features have less dimension than the log-spectral features), in which case a k-dependent small fraction of the identity matrix is added to regularize $\mathbf{H}_k$, so a unique solution of (11) is found. Since $\mathbf{H}_k$ is not diagonal, the bounded minimization in (11) can no longer be solved by Equation (2). Instead, it becomes a Constrained Least Square (CLSQ) problem, which does not have an analytical solution. Methods such as the MAP algorithm [24], primal active set methods [31], Multiplicative Updates (MU) [32], and imputation with PROSPECT features [33] have been proposed. But, their computational cost for large vocabulary speech recognizers with tens or hundreds of thousands of Gaussians becomes prohibitive. Below, the MC MDT imputation is proposed to significantly reduce the computational intensity to achieve an MDT recognizer with speed.

4. MC missing data imputation

In (11), Gaussian-based imputation is formulated as searching for the optimal clean speech vector within a *feasible* region, i.e., the (continuous) subspace which is spanned by the unreliable components and is bounded by the observation. This process can be approximated by evaluating each Gaussian on a list of feasible clean speech candidates and then selecting the candidate which maximizes the likelihood as the imputed value. This approximation is the basic idea behind the MC MDT imputation. For every Gaussian, the list of candidates is given by the imputation from a small set of CG. The Gaussians in the acoustic model, typically a large number, will be called BGs in the remainder of the article. The optimization for each BG in (11) is then approximated by selecting the $\hat{l}_{t,k}$ th clean speech candidate such that:

$$\hat{l}_{t,k} = \underset{l \in \Omega_k}{\arg\max}\, p\left(\tilde{\mathbf{x}}_{t,u,l}, \mathbf{y}_{t,r} \middle| k\right), \text{ i.e. } \hat{\mathbf{x}}_{t,u,k} \cong \tilde{\mathbf{x}}_{t,u,\hat{l}_{t,k}} \quad (12)$$

where Ω_k represents all the CGs which might generate suitable solutions for Gaussian k, and $\tilde{\mathbf{x}}_{t,u,l}$ is the clean speech estimate of the unreliable speech components obtained from CG l. The construction of Ω_k will be detailed in Section 4.3. Hence, in MC MDT, solving the CLSQ problem of BG k is replaced by L_k likelihood evaluations, where L_k is the cardinality of Ω_k. Whereas solving a large number of BG imputation problems is avoided, the task is shifted to the restricted set of CGs. Solving each of these problems requires a computational effort that is at least an order of magnitude greater than the evaluation of a Gaussian likelihood, so various approaches for the imputation with CGs are discussed below.

4.1. ML-imputation for CG

The imputed value from the CGs can be computed by iterative approaches such as Gradient Descent (GD) [33], MU [32], or MAP [24]. In GD, the gradient for iteration τ is

$$\mathbf{g}_{t,k}{}^{\tau} = \mathbf{H}_k(\mathbf{x}_{t,k}{}^{\tau-1} - \mathbf{C}^+\boldsymbol{\mu}_k)$$

where each negative component of $\mathbf{g}_{t,\,k}{}^{\tau}$, for which $\mathbf{x}_{t,\,k}$ is on the boundary $\mathbf{y}_t$, is zeroed and so is each reliable component of $\mathbf{g}_{t,\,k}{}^{\tau}$ in order to not violate the constraints. Since the cost function in Equation (10) is quadratic, the optimal step for iteration τ has an analytic expression:

$$step_{t,k}{}^{\tau} = \frac{-(\mathbf{g}_{t,k}{}^{\tau})'\mathbf{g}_{t,k}{}^{\tau}}{(\mathbf{g}_{t,k}{}^{\tau})'\mathbf{H}_k\mathbf{g}_{t,k}{}^{\tau}}\mathbf{g}_{t,k}{}^{\tau} \quad (13)$$

The step direction is maintained, but the step size is reduced such that the boundary constraints are not violated.

$$\mathbf{x}_{t,k}{}^{\tau} = \mathbf{x}_{t,k}{}^{\tau-1} + step_{t,k}{}^{\tau} \quad \mathbf{x}_{t,k}{}^{\tau} \le \mathbf{y}_t$$

To initialize the GD algorithm, the non-diagonal covariance structure is ignored, i.e., it starts from the solution in Equation (2).

We opt for GD rather than MU [32] or MAP [24] because it benefits from several advantages simultaneously: (i) the number of iterations required for practical convergence is smaller [33], (ii) the gradient computation (13) can be carried out from right to left such that only a small number of matrix-vector multiplications and vector operations are required (see Appendix), (iii) only the *constant* transformation matrix **C**, the observation, mean, and variances need to be copied to the cache memory of the CPU while other methods may require a larger memory access bandwidth, (iv) GD does not require square root operations like MU, hence the total number of arithmetic operations per iteration is less than that of MU (as shown in Table 1).

The computational effort is further reduced by using PROSPECT features together with GD, which is proposed in [33].

PROSPECT features are composed of two feature subset. The first are cepstral features of a low order D_c (e.g.,

Table 1 The expensive operations in the different methods for Gaussian-based MLE imputation with full covariance matrices

	Full precision matrix in cache	Number of iterations	Likelihood #multiplications	Step calculation #multiplications	Other
MAP [24]	Yes	6	$(D + 1)D$	$(D_{t,u} + 1)D_{t,u}$	No
MU [32]	Yes	5	$(D + 1)D$	$D_{t,u}D_{t,u} + 4D_{t,u}$	$D_{t,u}$ √ per step
GD + cepstral	No	2	$2D_mD + D_m + D$	$D_{t,u}(2D_m + 3) + D_mD + D_m + D$	No
GD + PROSPECT	No	2	$2(D_c + D)$	$3D_cD_{t,u} + 4D_cD + 5D + 3D_c$	No

D is the dimension of the log spectral features. D_m is the order of the cepstral coefficients in MFCC. $D_{t,u}$ is the number of unreliable components at frame t. D_c is the order of the cepstra in PROSPECT features. Typical values: $D = 22$, $D_m = 13$, $D_c = 4$. $D_{t,u}$ is about 16 on average as measured in [33].

$D_c = 4$), which models the rough shape of the spectrum at time t. This cepstral part is given by

$$\mathbf{v}_t^c = \mathbf{C}_c\mathbf{x}_t$$

where $\mathbf{C}_c$ denotes the reduced DCT matrix with orthonormal rows. The remaining details of the signal are captured by

$$\mathbf{v}_t^\perp = (\mathbf{I} - \mathbf{C}_c'\mathbf{C}_c)\mathbf{x}_t$$

which is termed the projection part because it is the orthogonal projection of $\mathbf{x}_t$ on the orthogonal complement of the subspace spanned by the rows of $\mathbf{C}_c$. The concatenation of the cepstral part and the projection part is referred to as PROjected SPECTral (PROSPECT) features:

$$\mathbf{v}_t = \begin{bmatrix} \mathbf{v}_t^c \\ \mathbf{v}_t^\perp \end{bmatrix} = \mathbf{R}\mathbf{x}_t$$

The PROSPECT transformation matrix is

$$\mathbf{R} = \begin{bmatrix} \mathbf{C}_c \\ (I - \mathbf{C}_c'\mathbf{C}_c) \end{bmatrix}$$

The likelihood of the kth Gaussian based on the PROSPECT features is formulated as

$$F(\mathbf{v}_t|k) = p(\mathbf{v}_t^c|k)p(\mathbf{v}_t^\perp|k)^\beta \tag{14}$$

where

$$p(\mathbf{v}_t^c|k) = \frac{1}{\sqrt{(2\pi)^{D_c}\,|\boldsymbol{\Sigma}_k^c|}} \exp\left(-\frac{1}{2}(\mathbf{v}_t^c - \boldsymbol{\mu}_k^c)'\boldsymbol{\Sigma}_k^{c^{-1}}(\mathbf{v}_t^c - \boldsymbol{\mu}_k^c)\right) \tag{15}$$

and

$$p(\mathbf{v}_t^\perp|k) = \frac{1}{\sqrt{(2\pi)^{D}\,|\boldsymbol{\Sigma}_k^\perp|}} \exp\left(-\frac{1}{2}(\mathbf{v}_t^\perp - \boldsymbol{\mu}_k^\perp)'\boldsymbol{\Sigma}_k^{\perp^{-1}}(\mathbf{v}_t^\perp - \boldsymbol{\mu}_k^\perp)\right) \tag{16}$$

where $\boldsymbol{\mu}_k^c$, $\boldsymbol{\Sigma}_k^c$, $\boldsymbol{\mu}_k^\perp$, and $\boldsymbol{\Sigma}_k^\perp$ are the means and covariance matrices of cepstral and projection part of PROSPECT Gaussian k, respectively. They are all estimated on data using the EM-algorithm and both $\boldsymbol{\Sigma}_k^c$ and $\boldsymbol{\Sigma}_k^\perp$ are diagonal. However, a diagonal $\boldsymbol{\Sigma}_k^\perp$ implies invalid independence assumptions in the spectral residual $\mathbf{v}_t^\perp$. Hence, the stream exponent β in (14) is introduced to reduce the impact of these assumptions. According to [33], a typical value of β is 0.5. Note that $F(\mathbf{v}_t|k)$ is not a strict PDF because it does not integrate to unity due to β, but we will still refer to it as the likelihood of Gaussian k. When substituting (15) and (16) in (14), the cost function of Gaussian k becomes

$$Q_k = (\mathbf{x}_t - \mathbf{R}'\boldsymbol{\mu}_k)'\left[\mathbf{C}_c'\boldsymbol{\Sigma}_k^{c^{-1}}\mathbf{C}_c + \beta(\mathbf{I} - \mathbf{C}_c'\mathbf{C}_c)\boldsymbol{\Sigma}_k^{\perp^{-1}}(\mathbf{I} - \mathbf{C}_c'\mathbf{C}_c)\right](\mathbf{x}_t - \mathbf{R}'\boldsymbol{\mu}_k) \tag{17}$$

where

$$\boldsymbol{\mu}_k = \begin{bmatrix} \boldsymbol{\mu}_k^c \\ \boldsymbol{\mu}_k^\perp \end{bmatrix}$$

The gradient computation and cost function evaluation now involve only multiplication of small matrices and vector additions, which is exploiting the CPU's cache memory more efficiently and reduces the computational effort in comparison to a cepstral (or LDA) model, as witnessed by Table 1. Refer to Appendix for details. The study [33] also shows that the PROSPECT model performs equally well as the cepstral model for a recognizer without MDT. Because of their better efficiency and comparable accuracy, the PROSPECT features are preferred for the CGs and the algorithm selected for minimizing (17) is GD. Since the CGs only serve to generate candidate spectra, there is no need for the CG and the BG to be expressed in the same feature domain. For example, in the experiments of Section 6, the BGs will be trained with the features generated by the Mutual Information-based Discriminant Analysis (MIDA) technique [34].

4.2. Training the CGs

Clustering methods for Gaussians can be categorized as model-based or data-driven. In the former methods, such as the popular K-means, the parameters of the

CGs are estimated from parameters of the BGs. In the latter methods, parameters of the CGs are estimated from training data. Model-based Gaussian clustering methods are not well suited to create the CGs in MC MDT, because they would involve a transformation between the domains in which CGs and BGs are expressed. For instance, MIDA CGs can be first evaluated using MIDA BGs and then converted into PROSPECT CGs. But this conversion involves a lossy transformation and hence performance cannot be guaranteed. Therefore, approaches driven by data are selected in this study.

In order to obtain the CGs from training data, a compact HMM is trained. The compact model shares its structure with the backend model containing the BGs in the sense that it uses the same phonetic decision tree (PDT) [35], but it has only M Gaussians which are shared among leaves of the PDT. Hence, every HMM state s will have an associated set of CGs as well as a set of BGs, denoted by $G_{\mathrm{CG}}(s)$ and $G_{\mathrm{BG}}(s)$, respectively. Typically, M is a few hundred and $M << K$, where K is the total number of BGs. These M Gaussians are used as the CGs and can be trained for any feature representation. The parameters to be trained are the PROSPECT means and covariance matrices of the CGs as well as the mixture weights $P_{\mathrm{CG}}(m|s)$. Before training the compact model, a state level segmentation is made using the Viterbi algorithm with the backend model, i.e., the segmentation specifies the alignment between the states and the frames of the training data. M BGs are randomly selected to initialize the CGs. However, since BGs and CGs may be expressed on different feature sets, a particular initialization of the CGs is required. Hereto, the M retained BGs are considered as a GMM with uniform weights. The posterior probabilities of the M BGs are calculated on the MIDA representation and are used in the first iteration of the EM algorithm, i.e., the BG posteriors are used to softly assign training samples to the CGs to initialize the mean, covariance and mixture weights. Subsequently, a standard EM training without altering the segmentation is performed using PROSPECT features. Consequently, each tied state is now modeled by a GMM with up to M components trained on PROSPECT features. Finally, every BG can be assigned to multiple CGs to form a soft clustering, as explained below.

4.3. Association between CGs and BGs

The association between the CGs and the BGs is based on the same segmentation used in Section 4.2. In this step, the likelihood of all the BGs belonging to state s at training frame t is calculated along the Viterbi path. The likelihood of the PROSPECT CGs belonging to s is calculated for the same frames. Then CG $\hat{m}_t$ and BG $\hat{k}_t$ are found by

$$\hat{m}_t = \underset{m \in G_{\mathrm{CG}}(s)}{\arg\max} P_{\mathrm{CG}}(m|s) F(\mathbf{Ry}_t|m)$$

and

$$\hat{k}_t = \underset{k \in G_{\mathrm{BG}}(s)}{\arg\max} P_{\mathrm{BG}}(k|s) p(\mathbf{Ey}_t|k)$$

where $F(\mathbf{Ry}_t|m)$ is calculated by Equation (14). $\mathbf{E}$ represents the linear transformation of the backend features. For every training frame of speech t, entry $(\hat{m}_t, \hat{k}_t)$ of the association matrix $\mathbf{\Phi}$ (as shown in Figure 2) is incremented by 1. After all training data are processed, the set Ω_k in Equation (12) is defined from the kth column of $\mathbf{\Phi}$ as those entries that are larger than the product of a pruning threshold θ_Φ and the maximum of the kth column. Moreover, if Ω_k contains more than $L_{\max}$ elements, only the $L_{\max}$ largest $\mathbf{\Phi}$-values are retained. The entries of $\mathbf{\Phi}$ that are not in Ω_k are subsequently set to zero.

The probability how often CG m is associated with any BGs is formulated by

$$P(m) = \frac{\sum_{k=1}^{K} \delta_{mk}}{\sum_{i=1}^{M} \sum_{k=1}^{K} \delta_{ik}} \quad \delta_{mk} = \begin{cases} 1 \ if \ \mathbf{\Phi}_{mk} > 0 \\ 0 \ if \ \mathbf{\Phi}_{mk} = 0 \end{cases} \tag{18}$$

and is used as the prior probability of a CG below.

4.4. Application of MC MDT in the recognizer

Figure 2 illustrates the process of calculating the likelihoods of the BGs. Since the CGs and the CG-BG association table are now available, it is also convenient to apply Gaussian selection together with the MC MDT. The motivation of Gaussian selection is that only a small (frame dependent) portion of Gaussians dominate the likelihoods of the HMM states, and are therefore worth evaluating. However, since the likelihood computation in an MDT system involves the imputation of the unknown data, many conventional methods do not apply readily. The proposed approach proceeds as follows. The recognizer first evaluates all the PROSPECT CGs using GD. Only the BGs assigned (as determined by Ω_k) to sufficiently likely CGs will be calculated, while the others will be ignored. The prior probabilities (18) and the resulting likelihoods of CGs are used to calculate the corresponding posterior probabilities. The posterior probabilities are sorted in descending order and then truncated to the length such that a large fraction ρ (e.g. $\rho = 0.95$) of the posterior probability mass is included, i.e., the number of CGs kept is the smallest L_s such that

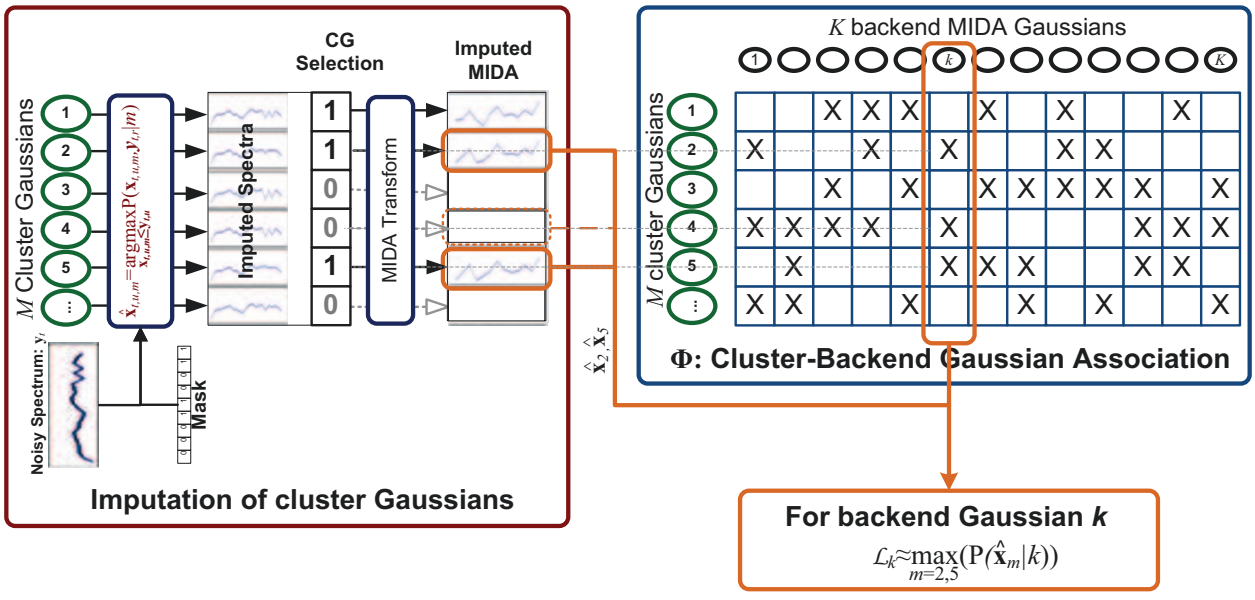

Figure 2 The process of calculating the likelihoods of the BGs in the Gaussian-based MC MDT system. For each of the *M* CG (left), a clean speech spectrum is imputed using MDT and solutions with high posterior probability (CG selection table) are transformed to the feature representation of the backend (imputed MIDA). BGs (right) are evaluated at the candidates listed in the association table (crosses) and the highest likelihood is retained (bottom).

$$\sum_{m=1}^{L_s} P\left(m \left| \hat{\mathbf{x}}_{t,u,m}, \mathbf{y}_{t,r}\right.\right) > \rho \sum_{i=1}^{M} P\left(i \left| \hat{\mathbf{x}}_{t,u,i}, \mathbf{y}_{t,r}\right.\right) \tag{19}$$

where $\hat{\mathbf{x}}_{t,u,m}$ is the imputed value from CG m. The CG are reordered such that

$$P\left(m \left| \hat{\mathbf{x}}_{t,u,m}, \mathbf{y}_{t,r}\right.\right) \geq P\left(m+1 \left| \hat{\mathbf{x}}_{t,u,m+1}, \mathbf{y}_{t,r}\right.\right)$$

and

$$P\left(m \left| \hat{\mathbf{x}}_{t,u,m}, \mathbf{y}_{t,r}\right.\right) \cong \frac{p\left(\hat{\mathbf{x}}_{t,u,m}, \mathbf{y}_{t,r} \left| m\right.\right)^{\alpha} P(m)}{\sum_{i=1}^{M} p\left(\hat{\mathbf{x}}_{t,u,i}, \mathbf{y}_{t,r} \left| i\right.\right)^{\alpha} P(i)} \tag{20}$$

The exponent α is introduced to compensate for unmodeled correlations among the features and will indirectly control the number of selected BGs. A typical value of α is 0.4, which led to a reasonable trade-off between the number of selected BG and recognition accuracy on the development dataset used in [36]. $P(m|\hat{\mathbf{x}}_{t,u,m}, \mathbf{y}_{t,r})$ denotes the posterior probability of CG m based on its imputation. In Figure 2, the CGs labeled with "1" in the CG selection table are selected at frame t. Only the imputed clean spectra resulting from the selected CGs are transformed into the MIDA domain and maintained as possible candidates for BG likelihood evaluation.

When calculating the likelihood of a particular BG k, the MC MDT recognizer looks up the kth column of the CG-BG association table $\mathbf{\Phi}$ to find the candidate list. Notice that some of the associated CGs may have been pruned by criterion (19) and are removed from the list. The recognizer calculates the likelihoods of the BG for the candidates of imputed clean speech and selects the maximum as the likelihood of that BG. If the candidate list is empty, the BG is assigned a likelihood of zero. On average, the number of multiplications involved per BG is reduced to $2\bar{L}D_m$, where $\bar{L}$ is the average number of CGs associated to a BG and D_m is the dimension of MIDA features. The resulting likelihoods of the BGs are used to calculate the state output PDFs, which are then processed by the decoder.

5. Selection of CG

The MC MDT system can be further sped up by applying Gaussian selection on the CGs. Though $M << K$, the evaluation of a CG is still an order of magnitude more expensive than the evaluation of a BG. Thus, only the likely CGs are selected to impute the candidate clean speech. Existing methods of Gaussian selection can be classified as axis indexing-based methods [37,38] and VQ-based methods [39,40]. The former quickly locates the likely regions based on the observation, then selects the Gaussians in the likely regions [38] or removes the Gaussians in the unlikely regions [37]. But in MDT systems, it is not straightforward to determine which regions in the feature space are likely, because some of

the components of the observation are missing. On the contrary, VQ-based methods suit the MC MDT system well. Cluster-of-Cluster Gaussians (CCG) in the PROSPECT domain are now established. The MC MDT recognizer will select the CGs based on the likelihoods resulting from the imputation of CCGs, i.e., an additional layer of Gaussian selection is provided. Consequently, it reduces the number of CG CLSQ problems to be solved. Clustering of the CGs is a prerequisite for the VQ-based Gaussian selection. In this study, we apply the soft K-Means algorithm to generate the CCGs. Since the CCGs and the CGs are expressed in the same domain (PROSPECT features in our example), a model-based approach is feasible and preferred here.

5.1. Soft *K*-means clustering

The following pseudo code summarizes the steps to obtain the CCGs. A single cluster is first calculated using all the CGs. The number of CCGs then grows incrementally from 1 to N to avoid suboptimal clustering as much as possible.

1. Set the number of CG n to 1 and compute a single CCG from all CGs.
2. While $n < N$
   - 2a. Find CCG $\hat{j}$ with the maximum mean WSKLD
   - 2b. Split CCG $\hat{j}$ into two and increment n
   - 2c. For iteration τ from 1 to T
   - 2c-1. For CCG i, i from 1 to n
   - 2c-1-1. For CG m, m from 1 to M
     Calculate the weight by which CG m updates CCG i, $\hat{g}(i, m)$
   - 2c-1-2. Given $\hat{g}(i, m)$, update $\boldsymbol{\mu}_i$, $\boldsymbol{\Sigma}_i$ iteratively

The distance metric between Gaussians and the computation of the CCGs from its member CGs are two crucial components for every step listed in the above pseudo code. The distance metric is Weighted Kullback-Leibler Divergence (WSKLD) in step 2a and is explained in Section 5.2. The parameter estimation algorithms in steps 1, 2c-1-1, and 2-c-1-2 are described in Section 5.3. Step 2b is described in Section 5.4.

5.2. Distance metric between PROSPECT Gaussians

The symmetric Kullback-Leibler Divergence (SKLD) is commonly used to measure the distance between CCG n and CG m:

$$\mathrm{SKLD}(n, m) = \frac{1}{2}\mathrm{trace}\left((\boldsymbol{\Sigma}_n{}^{-1} + \boldsymbol{\Sigma}_m{}^{-1})(\boldsymbol{\mu}_n - \boldsymbol{\mu}_m)(\boldsymbol{\mu}_n - \boldsymbol{\mu}_m)' + \boldsymbol{\Sigma}_n{}^{-1}\boldsymbol{\Sigma}_m + \boldsymbol{\Sigma}_n\boldsymbol{\Sigma}_m{}^{-1} - 2\mathbf{I}\right)$$

The application of SKLD to (14) requires some care: the stream exponent β in the likelihood model for PROSPECT features makes it an improper distribution, requiring renormalization such that it integrates to unity. Second, it was found that SKLD overweighs differences in the projection part of the PROSPECT Gaussians. Therefore, in [41], further simplifications were proposed and experimentally verified leading to the WSKLD as a clustering metric for multi-stream features:

$$\mathrm{WSKLD}(n, m) = \sum_{j=1}^{N_{strm}} \beta_j \,\mathrm{SKLD}_j(n, m)$$

where β_j is the exponent of stream j, SKLD_j is the symmetric KLD computed on the features of stream j only and N_{strm} is the total number of streams. In this study, N_{strm} is 6 because the PROSPECT features contain static, velocity and acceleration stream of both cepstral and projection parts.

5.3. Parameter estimation of CCGs

Following the K-Means algorithm in [42], the cost function to be minimized for clustering is

$$\mathrm{Q}_{K-Means} = \sum_{m=1}^{M}\left(\sum_{n=1}^{N} \mathrm{g}(n,m)\mathrm{WSKLD}(n, m) + \gamma \sum_{n=1}^{N} \mathrm{g}(n,m)\log\frac{1}{\mathrm{g}(n, m)}\right) \quad (21)$$

where γ controls the stiffness of the clustering and $g(n, m)$ are unknown clustering weights. The parameters to be updated iteratively are

$$[\hat{\boldsymbol{\mu}}_n, \hat{\boldsymbol{\Sigma}}_n, \hat{\mathrm{g}}(n, m)] = \underset{\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n \sum_{n=1}^{N} g(n,m)=1}{\arg\min} (\mathrm{Q}_{K-Means})$$

In each iteration, the first step is to obtain the optimal weight by which CG m affects CCG n as

$$\hat{\mathrm{g}}(n, m) = \frac{\exp(-\mathrm{WSKLD}(n, m)/\gamma)}{\sum_{i=1}^{N}\exp(-\mathrm{WSKLD}(i, m)/\gamma)}$$

The second step is to find the optimal values of mean and covariance of each CG given the weights. The estimation of means and covariance matrices of the CCGs is based on the approach in [43], where a method for finding the centroid of a set of Gaussians is derived. The centroid is the CCG that minimizes the sum of the WSKLD to all CGs. Here, we extend the results of [43] by modifying the cost function to (21). The mean of a CCG is thereby estimated as

$$\hat{\boldsymbol{\mu}}_n = \left[\sum_{m=1}^{M}\hat{\mathrm{g}}(n, m)(\boldsymbol{\Sigma}_m^{-1} + \boldsymbol{\Sigma}_n^{-1})\right]^{-1}\left[\sum_{m=1}^{M}\hat{\mathrm{g}}(n, m)(\boldsymbol{\Sigma}_m^{-1} + \boldsymbol{\Sigma}_n^{-1})\boldsymbol{\mu}_m\right] \quad (22)$$

Matrix $\mathbf{Z}$ is constructed to facilitate the re-estimation of the covariance matrix of the CCGs

$$\mathbf{Z} = \begin{bmatrix} 0 & \mathbf{A}_1 \\ \mathbf{A}_2 & 0 \end{bmatrix}$$

where

$$\mathbf{A}_1 = \sum_{m=1}^{M} \hat{\mathrm{g}}(n,m)\left[(\boldsymbol{\mu}_m - \hat{\boldsymbol{\mu}}_n)(\boldsymbol{\mu}_m - \hat{\boldsymbol{\mu}}_n)' + \Sigma_m\right]$$

and

$$\mathbf{A}_2 = \sum_{m=1}^{M} \hat{\mathrm{g}}(n,m)\Sigma_m^{-1}$$

By construction, $\mathbf{Z}$ has D_P positive and D_P symmetrically negative eigenvalues, where D_P is the dimension of PROSPECT features. A $2D_P$-by-D_P matrix $\mathbf{V}$ is constructed whose columns are the D_P eigenvectors corresponding to the positive eigenvalues. $\mathbf{V}$ is partitioned in its upper halve $\mathbf{U}$ and lower halve $\mathbf{W}$:

$$\mathbf{V} = \begin{bmatrix} \mathbf{U} \\ \mathbf{W} \end{bmatrix}$$

$$\hat{\boldsymbol{\Sigma}}_n = \mathbf{U}\mathbf{W}^{-1} \tag{23}$$

Like in [43], $\hat{\boldsymbol{\Sigma}}_n$ is constrained to be diagonal during clustering. It can be seen from Equations (22) and (23) that the procedure of estimating the CCGs given the weights $\hat{g}(n,m)$ is iterative. The calculation of the mean depends on the previously calculated covariance and vice versa. The exit criterion is the convergence of the cost function defined in Equation (21).

In step 1 of the pseudo code from Section 5.1, a single CCG is initialized by averaging the means and covariance matrices of the entire set of CGs. The parameters of the single CCG are then updated using Equations (22) and (23) for several iterations. Splitting a CCG and re-estimation of all CCGs are carried out iteratively till N CCGs are obtained, as explained below.

5.4. Splitting a CCG

In each iteration, the CCG with the maximum within-cluster mean WSKLD is found

$$\hat{j} = \arg\max_{j=1\ldots n} \frac{\sum_{m=1}^{M} \hat{\mathrm{g}}(j,m)\mathrm{WSKLD}(j,m)}{\sum_{m=1}^{M} \hat{\mathrm{g}}(j,m)}$$

Principal component analysis is applied on the covariance matrix $\Sigma_{\hat{j}}$ to find the first principal eigenvector $\boldsymbol{e}_1$ and eigenvalue, λ_1. If the number of CCGs in the current iteration is n, CCG $\hat{j}$ is split into two Gaussians with the means and covariance matrices:

$$\boldsymbol{\mu}_{n+1} \leftarrow \boldsymbol{\mu}_{\hat{j}} + \xi\sqrt{\lambda_1}\mathbf{e}_1$$

$$\boldsymbol{\mu}_{\hat{j}} \leftarrow \boldsymbol{\mu}_{\hat{j}} - \xi\sqrt{\lambda_1}\mathbf{e}_1$$

$$\boldsymbol{\Sigma}_{n+1} = \boldsymbol{\Sigma}_j = \boldsymbol{\Sigma}_{\hat{j}}$$

where ξ is the disturbing rate. The WSKLDs of all the M CGs to the newly created CCGs is then calculated. Each weight $\hat{g}(\hat{j},m)$ is also split into two according to the WSKLDs:

$$\hat{\mathrm{g}}(n+1,m) \leftarrow \frac{\exp(-\mathrm{WSKLD}(n+1,m)/\gamma)\hat{\mathrm{g}}(\hat{j},m)}{\exp(-\mathrm{WSKLD}(n+1,m)/\gamma) + \exp(-\mathrm{WSKLD}(\hat{j},m)/\gamma)}$$

$$\hat{\mathrm{g}}(\hat{j},m) \leftarrow \hat{\mathrm{g}}(\hat{j},m) - \hat{\mathrm{g}}(n+1,m)$$

The parameters of CCG $\hat{j}$ and CCG $n+1$ are then re-estimated using Equations (22) and (23) with fixed number (e.g., 3) of iterations. The means and covariance matrices of CCG 1 to $n+1$ are subsequently updated until convergence of the global cost function (21).

Finally, when n reaches N, an N by M CCG-CG table of exponentiated negative WSKLD is calculated. This table plays the same role as the association table in Section 4.3. The same schemes as in Section 4.3 are used to truncate the table. Also, the same schemes as in Section 4.4 are used to select likely CGs, thus avoiding solving CLSQ problems whose solutions are unlikely to survive pruning criterion (19).

6. Experiments

To show the effectiveness of the proposed approaches, a large vocabulary speech recognizer was modified accordingly and experiments on the noisy dictation task AURORA-4 were run. In Section 6.1, we describe the training and test datasets and further details on the required acoustic models. Section 6.2 explains various components of the MDT recognizer. Section 6.3 outlines two baseline systems: first, a non-MDT system serving as the *speed* baseline for our MC MDT experiments and second a backend MDT system serving as the *accuracy* baseline for our MC MDT experiments. In Section 6.4, some MC MDT variants are first analyzed and subsequently compared with the cluster-based reconstruction described in Section 6.5. Section 6.6 evaluates MC MDT systems where the CGs are expressed in either the cepstral domain or the log-spectral domain. All testing results are summarized in Tables 2, 3, 4, and 5.

Table 2 WER of the MC MDT experiments on test set 1-7 (Close-talking microphone)

	WER (%)							
	1. Clean	2. Car	3. Babble	4. Restau.	5. Street	6. Airport	7. Train	Average
1. No MDT	6.82	12.98	32.62	40.95	38.50	32.51	38.37	28.96 ± 0.5
2. BE PROSPECT MDT	6.33	10.89	20.14	28.38	25.42	20.57	25.14	19.55 ± 0.4
3. GB GD MC MDT	6.28	10.95	20.08	29.76	24.98	20.18	25.28	19.64 ± 0.4
4. GB MAP MC MDT	6.00	10.31	20.90	29.12	24.86	19.73	25.89	19.55 ± 0.4
5. GB GD MC MDT + CGS	6.16	10.93	20.36	29.48	25.72	19.91	25.13	19.53 ± 0.4
6. GD CLBR	6.18	11.96	22.90	31.22	26.60	21.74	29.03	21.38 ± 0.4
7. MAP CLBR	6.28	11.81	21.69	31.44	27.63	21.74	28.51	21.30 ± 0.4
8. SB MC MDT	6.67	11.25	20.25	28.90	24.49	19.67	25.20	19.49 ± 0.4
9. Spectral MC MDT	6.41	10.46	21.63	29.68	25.89	20.66	25.85	20.08 ± 0.4
10. Cepstral MC MDT	6.44	11.04	19.90	29.46	25.61	19.86	26.53	19.83 ± 0.4

GB, Gaussian-based; SB, State-based; CGS, CG selection; CLBR, cluster-based reconstruction; BE, Backend.

6.1. Data and models

6.1.1. AURORA-4

Speech recognition experiments were conducted on the AURORA-4 database [44], a large vocabulary task that is derived from the WSJ0 Wall Street Journal 5k-word dictation corpus. A bigram language model for a 5k-word closed vocabulary is provided by Lincoln Laboratory.

For training, only clean-condition data sampled at 16 kHz were used, consisting of 7,138 utterances from 83 speakers, which amounts to 14 h of speech data. All recordings are made with the close talking microphone and no noise is added.

The test database is composed of 330 read sentences (5,353 words) from 8 different speakers. Fourteen different versions of this set are created. The first dataset is clean and is recorded with the same close-talk microphone as used while recording the training data. It is artificially corrupted by adding six types of noise to establish datasets 2-7: car (set 2), babble (set 3), restaurant (set 4), street (set 5), airport (set 6), and train (set 7). Set 8 is recorded with far-talk microphones. Test sets 9-14 are created by artificially adding the same six types of noise as used for generating sets 2-7. Each test set contains 330 utterances and has an SNR that ranges from 5 to 15 dB.

6.1.2. Training backend acoustic model

The design of the front-end as well as the backend acoustic model is based on prior study [23,33,37] which obtained competitive accuracies on clean speech and good robustness in an MDT configuration.

The signal power spectrum is calculated with a 32-ms Hamming window and a 10-ms window shift and is integrated using a 22-channel MEL-scaled triangular filter bank with lowest frequency centered at 140 Hz to increase the robustness to low-frequency noises. Since all frequencies above 7 kHz of the AURORA-4 data are filtered out, the last band is centered at 5800 Hz. The 22 log-spectral coefficients are mean-normalized and the first- and second-order time derivatives are appended to result in 66-dimensional spectral features.

To train the backend acoustic model, the normalized spectral features are transformed into 39-dimensional MIDA features, which are improved LDA features

Table 3 WER of the MC MDT experiments on test set 8-14 (Far-talking microphone)

	WER (%)							
	8. Clean	9. Car	10. Babble	11. Restau	12. Street	13. Airport	14. Train	Average
1. No MDT	21.35	30.51	47.71	53.26	55.52	46.95	55.41	44.39 ± 0.5
2. BE PROSPECT MDT	15.93	23.02	36.39	41.29	42.14	34.65	41.32	33.54 ± 0.5
3. GB GD MC MDT	15.93	23.63	37.15	42.27	41.43	35.29	40.91	33.80 ± 0.5
4. GB MAP MC MDT	15.51	23.58	37.17	42.13	41.99	35.00	41.55	33.84 ± 0.5
5. GB GD MC MDT + CGS	15.86	23.96	37.43	42.49	41.58	35.15	40.83	33.90 ± 0.5
6. GD CLBR	19.39	27.93	40.54	45.75	48.14	38.88	45.45	38.01 ± 0.5
7. MAP CLBR	17.86	27.54	43.96	48.10	48.10	39.73	46.98	38.90 ± 0.5
8. SB MC MDT	15.88	23.01	37.23	41.93	41.39	34.99	41.71	33.73 ± 0.5
9. Spectral MC MDT	15.77	24.42	38.74	44.57	42.42	36.17	42.54	34.95 ± 0.5
10. Cepstral MC MDT	16.05	23.04	36.57	42.37	41.74	35.07	41.84	33.62 ± 0.5

GB, Gaussian-based; SB, State-based; CGS, CG selection; CLBR, cluster-based reconstruction; BE, Backend.

Table 4 Average CPU time breakdown over test set 2-7

	CPU time (ms/frame)					BG calculated (%)
	Mask estimation	BG Calculation	CG Evaluation	Beam Search	Total	
1. No MDT	0.0	3.5	0.2[a]	24.4	28.1	27
2. BE PROSPECT MDT	1.1	133.5	0.0	18.5	153.1	100
3. GB GD MC MDT	1.1	2.8	4.3	20.0	28.2	11
4. GB MAP MC MDT	1.1	3.3	7.6	22.2	34.2	13
5. GB GD MC MDT+CGS	1.1	2.4	1.2	19.1	23.8	9
6. GD CLBR	1.1	1.3	4.3	19.3	26.0	11
7. MAP CLBR	1.1	1.2	10.6[b]	18.5	31.4	9
8. SB MC MDT	1.1	30.2	4.6	20.1	56.0	11
9. Spectral MC MDT	1.1	10.4	0.5	23.7	35.7	38
10.Cepstral MC MDT	1.1	2.8	7.0	19.0	29.9	10.5

[a]In the case of No MDT, the time for CG evaluation is replaced by FRoG. [b]In cluster-based reconstruction with MAP, the time of CG evaluation includes the calculation of the likelihood for 500 PROSPECT CGs, which is equivalent to evaluation of 1000 MIDA Gaussians.
GB, Gaussian-based; SB, State-based; CGS, CG selection; CLBR, Cluster-based reconstruction; BE, Backend.

leading to decorrelation and diagonalization of the mixture components [34]. It has only half the dimension of PROSPECT features (see Section 6.1.3), hence leading to a significant effort reduction in the likelihood calculation of the BGs and showing better accuracy than MFCC.

The acoustic model uses cross-word and context-dependent triphones. The HMM for each triphone contains three states. A PDT defines 4091 tied states, or senones, which in their turn share 21,037 BGs. The output probability of each state is a mixture of 190 BGs on average and each Gaussian is shared among 45 different tied states.

6.1.3. Training CGs and CCGs

The compact acoustic model containing the CGs is trained with the same training data by following the training procedure outlined in Section 4.2. Here, the state-level segmentation of the training data is obtained by forced alignment using the backend MIDA model of Section 6.1.2. The normalized static log-spectral features and the dynamic features are transformed into PROSPECT features with D_c = 4, i.e., for each stream in the features, four cepstral coefficients are kept, and D = 22 projection coefficients are appended. Consequently, the PROSPECT features including delta's have 78 dimensions. An earlier experiment on AURORA-4 showed that MC MDT with 500 to 900 CGs yields a reasonable trade-off between recognition time and accuracy. Therefore, we use 500 CGs in the following experiments. The association table $\mathbf{\Phi}$ is built on the same training data. The maximum number of CGs associated with a particular BG, $L_{\max}$, is 5. An earlier experiment on the Flemish Speecon and SpeechDat Car data [36] showed that increasing $L_{\max}$ beyond 5 only leads to more computation without increasing the recognition accuracy. The average number of CGs associated with a particular BG, $\bar{L}$, is 3.6.

Fifty CCGs are obtained by clustering the 500 CGs using the procedure from Section 5. The maximum

Table 5 CPU time breakdown for the clean condition (test set 1)

	CPU time (ms/frame)					BG calculated (%)
	Mask estimation	BG calculation	CG evaluation	Beam search	Total	
1. No MDT	0.0	1.9	0.2[a]	5.8	7.9	14
2. BE PROSPECT MDT	1.1	58.9	0.0	5.8	65.8	100
3. GB GD MC MDT	1.1	1.7	3.4	4.8	11.0	5
4. GB MAP MC MDT	1.1	2.2	6.2	6.6	16.1	6
5. GB GD MC MDT + CGS	1.1	1.7	0.7	5.6	9.1	4
6. GD CLBR	1.1	0.8	3.4	5.9	11.2	5
7. MAP CLBR	1.1	0.8	6.9[b]	7.2	16.0	5
8. SB MC MDT	1.1	16.9	3.8	6.0	27.8	5
9. Spectral MC MDT	1.1	6.5	0.5	6.5	14.6	21
10. Cepstral MC MDT	1.1	1.7	4.4	5.2	12.4	5

[a]In the case of No MDT, the time for CG evaluation is replaced by FRoG. [b]In cluster-based reconstruction with MAP, the time of CG evaluation include the calculation of likelihood for 500 PROSPECT CGs, which is equivalent to evaluation of 1000 MIDA Gaussians.
GB, Gaussian-based; SB, State-based; CGS, CG selection; CLBR, Cluster-based reconstruction; BE, Backend.

number of CCGs associated with a CG is 5 and the average number is 3.6. In previous experiments on Gaussian clustering, we have found a γ of 0.3 in Equation (21) to be a good choice, which we have maintained in these experiments.

6.1.4. Training spectral CGs for the cluster-based reconstruction with MAP

In order to accomplish the experiments of the MAP cluster-based reconstruction for comparison, a mixture of 500 Gaussians with full covariance is trained as well on the spectral data using EM on the same segmentation. As proposed in [24], the initial iterations use a diagonal covariance model that serves to make the posterior probability calculation (6) feasible. Only in the last EM-iteration, the full covariance matrices are estimated for application in Equation (5).

6.1.5. Training backend PROSPECT model

In order to show the speed improvement of MC MDT over a full MDT system [45], i.e., where the CLSQ problem (11) is solved per Gaussian with GD, an acoustic model with Gaussians estimated on PROSPECT features is required. The model has 21,037 PROSPECT Gaussians which are obtained by Single Pass Retraining (SPR) [46] of the acoustic model with MIDA features. The inputs of the SPR are the MIDA features, PROSPECT features, and the MIDA model described above. The MIDA model is used to compute the posterior probabilities of every Gaussian over the training data, which are subsequently combined with the PROSPECT feature observations to estimate their GMM weights, means and diagonal covariance matrices.

6.2. Recognizer

6.2.1. Handling convolutional noise

Besides additive noise, the MDT recognizer also handles convolutional noise by the channel compensation technique described in [23], which maximizes the likelihood of the recognized speech on the backend model. To make the implementation tractable, only the contribution of the single Gaussian that gives the largest contribution to the state likelihood (the *dominating* BG) is taken into account. However, unlike in [23], the current approach computes only approximate solution for BGs which is expressed in a different feature domain. Therefore, each dominating BG is replaced by the PROSPECT CG with the largest Φ-value so the maximum likelihood channel estimation of [23] can be readily applied. The channel estimate is subtracted from the observed log-spectra and hence the CCG, CG, and BG models are all compensated for convolutional distortions.

6.2.2. Mask estimation

The missing data detector used is the method described in [23] which integrates harmonicity and SNR with a speech model based on vector quantization. At each frame, the best match between a harmonic decomposition of noisy speech and a codebook describing the harmonic decomposition of clean speech is found. VQ mask estimation requires a speech and silence codebook which are trained with a randomly selected subset of the clean training data in Section 6.1.1. The codebook contains 520 codewords which are updated using the channel estimation of Section 6.2.1 during recognition.

6.2.3. Test configuration

The decoding consists of a time-synchronous beam search algorithm as described in [47]. The recognizer was launched on a PC installed with Dual Core AMD Opteron 280 2.4 GHz processors with a cache size of 1 MB. Only one processor core is activated. The MDT imputation is only applied to the static stream, while the first- and second-order time derivatives are uncompensated. The Word Error Rates (WER) are calculated for all the experiments. Meanwhile, the CPU time is measured. Tables 2 and 3 list the WERs of the experiments over the 14 types of environmental noises. Tables 4 and 5 contain the timing measurements for the BG and CG evaluation, for beam search as well as the end-to-end timing information (column "TOTAL") of the recognizer under noisy and clean condition, respectively. The timing measurements are achieved by starting and stopping (precise) timers frame-synchronously at the entry and exit of each of the different processing steps: front-end processing, CG imputation, candidate evaluation for all BGs, and beam search. The total time is then obtained by dividing the accumulated timings by the number of processed frames over several utterances.

6.3. Baselines

6.3.1. Recognition without MDT

The system that does not make use of MDT is provided as a baseline in terms of recognition time such that we can measure the computational cost of the robustness obtained from the MDT systems. The acoustic model is the backend HMM containing 21,037 MIDA Gaussians described in Section 6.1.2. An axis indexing-based Gaussian selection method, Fast Removal of Gaussians (FRoG) [37] is used. The testing results are shown in the first rows of Tables 2, 3, 4 and -5. The default FRoG Gaussian pruning setting works well on clean speech and results in only about 5% of Gaussians being evaluated. However, we noticed a performance degradation due to Gaussian pruning on noisy speech. Therefore, the FRoG Gaussian pruning settings were adjusted on the noisy test data such that the accuracy was not degraded more than 2% compared to no pruning, requiring 27% of Gaussians to be kept. Notice that this procedure yields an optimistic speed estimate for this baseline, as tuning on an independent development set would require some safety margin as well. Notice that

this non-MDT system produces higher WER than the MDT systems under the clean condition (test set 1), as shown in Table 2. This is mainly due to the non-MDT system using spectral mean normalization to reduce the channel effects, while the MDT systems use the more sophisticated MLE-based channel update as described in Section 6.2.1.

6.3.2. Backend PROSPECT imputation

This setup is the most refined previously published version of our MDT system [23] and serves as a baseline in term of recognition accuracy such that we can measure the accuracy cost of the proposed speed improvements. Two iterations of GD are found to be enough for the convergence in terms of recognition accuracy, hence are applied for all the 21,037 PROSPECT Gaussians. This system runs at 15 times real time in noisy condition and 6.6 times real time in clean condition as shown in row 2 of Tables 4 and 5, respectively. However, the accuracy benefits of MDT can be clearly seen in contrast to the non-MDT system in Tables 2 and 3.

6.4. MC MDT

6.4.1. Gaussian-based MC MDT with GD

The Gaussian-based MC MDT system is an instance of the concepts outlined in Section 4. Two iterations of GD are applied for all the 500 CGs. The posterior probability-based BG selection described in Section 4.4 is applied. α was tuned with an isolated word-recognition experiment of MC-MDT on the Speecon and the SpeechDat Car databases used in [36], which we hence regard as development data for this article. The tuning experiment shows that a good trade-off between accuracy and BG evaluation effort is obtained at $\alpha = 0.4$, but that it does not critically affect the recognition accuracy.

Compared to the backend PROSPECT MDT system, i.e., row 2 versus row 3 in Tables 2, 3, 4, and 5, the Gaussian-based MC MDT yields a comparable WER, while it uses less than 20% of the CPU time over the entire test set.

It is remarkable that the Gaussian-based MC MDT works as fast as the non-MDT recognizer with the same backend acoustic model under noisy conditions (row 1 versus row 3 of Table 4). The MC MDT spends time in evaluating CGs, but its decoding time is reduced by 4 ms per frame. Faster decoding on corrupted data is actually a common benefit from MDT imputation as shown in Table 4. In non-MDT systems, the mismatch between data and model results in a lower likelihood for the ground truth hypothesis and also causes many hypotheses to yield a similar score, so pruning is not effective and the decoder slows down. In the MDT system, noise addition also slows the recognizer down, but through a different mechanism. Thanks to the imputation process in MDT systems, the likelihood of the ground truth hypothesis will not deteriorate. The likelihood of alternative hypotheses will also increase, but because they do not fit the data well, their imputation benefit is not that strong. Apparently, a significant likelihood gap is maintained among the hypothesis, causing pruning in MDT systems to be more effective than in non-MDT systems. The effort spent in evaluating CGs is recovered in the search.

The increase in the likelihood of alternative hypotheses in the MDT system under noisy conditions also causes the MC MDT system with GD to run about 2.5 times slower than under clean conditions (row 3 of Table 4 versus row 3 of Table 5). As the data get noisier, the imputation becomes less constrained, since the number of unreliable components increases and the bounds outlined in Section 2.1 become less strict. Hence, the dynamic range of the BG likelihoods will decrease, such that the hypothesis likelihoods will show smaller differences, causing pruning to be less effective. Additionally, the system is slowed down with increasing noise levels because more spectro-temporal regions are labeled as unreliable and the complexity of imputation for CGs and CCGs increases.

Some common advantages of MDT are revealed by the results shown in Tables 2 and 3. All the experiments with MDT produce lower WERs than the non-MDT system over the corresponding noise types, as well as in the clean condition. Especially for the non-stationary noise types, namely, set 3-7 and 10-14, the benefit from MDT is more significant.

Though MDT systems show an advantage in both the close-talk and the far-talk test sets, the performance is greatly degraded in the latter condition, because the channel compensation technique of Section 6.2.1 is restricted to the estimation of a log-spectral offset vector, which can only compensate for convolutional effects with a short impulse response. However, the fact that the backend PROSPECT MDT and the MC MDT system perform equally on this test set confirms that the modification to estimate the channel on CGs rather than on BGs (see Section 6.2.1) works.

6.4.2. Gaussian-based MC MDT with MAP

This experiment is conducted to compare GD with MAP as a solver for the imputation problems. The full precision matrix of the CGs in Equation (17), as required for MAP, is pre-calculated. The same BG selection as in the previous section is applied. Six iterations are found to be enough for the convergence in terms of WER, and therefore applied for the CGs. Comparing row 3 and 4 of Tables 2 and 3 reveals that the MAP solver performs equally robust as the MC MDT using GD. But as shown in Tables 4 and 5, it is slower, especially in evaluating CGs due to more iterations and copying full precision matrices from the main memory to the cache memory of the CPU.

6.4.3. Gaussian-based MC MDT with CG selection

The CG selection introduced in Section 5 is added to the Gaussian-based MC MDT system in Section 6.4.1. In addition to the 50 PROSPECT imputation operations for the CCGs, about 106 PROSPECT imputation operations for the CGs are observed per frame of 10 ms. Therefore, the number of CLSQ problems solved is less than one-third of that of MC MDT without CG selection. Two iterations of GD are applied on the imputation of CCGs. The implementation of Gaussian-based MC MDT plus CG selection does not harm the recognition accuracy but consumes less CPU time in comparison with the Gaussian-based MC MDT system (compare row 3 with 5 in Tables 2, 3, 4, and 5).

6.4.4. State-based MC MDT

The imputed values of the Gaussian-based MC MDT described in Section 6.4.1 can also be used to perform a state-based MC MDT, where the imputation for state s is the linear combination of the imputed values from the BGs included in the GMM of that state.

$$\hat{\mathbf{x}}_{t,u,s} = \sum_{k \in G_{BG}(s)} P\left(k \,|\hat{\mathbf{x}}_{t,u,k}, \mathbf{y}_{t,r}, s\right) \hat{\mathbf{x}}_{t,u,k} \tag{24}$$

where $G_{\mathrm{BG}}(s)$ represents all the Gaussians belonging to the GMM of state s, and

$$P\left(k \,|\hat{\mathbf{x}}_{t,u,k}, \mathbf{y}_{t,r}, s\right) = \frac{p\left(\hat{\mathbf{x}}_{t,u,k}, \mathbf{y}_{t,r} | \, k\right) P(k|s)}{\sum_{j \in G_{BG}(s)} p\left(\hat{\mathbf{x}}_{t,u,j}, \mathbf{y}_{t,r} | \, j\right) P(j|s)} \tag{25}$$

The BG selection from Section 4.4 is also activated for this experiment, so only the selected BGs are actually involved in the imputation formulae. Each BG is shared among about 45 states and is therefore evaluated at multiple imputed spectra from its owner states. Hence, for this state-based MC MDT experiment, every selected BG is evaluated at 45 states-based imputed spectra as $\hat{\mathbf{x}}_{t,u,s}$ in Equation (24). The MC-based likelihood estimation of each BG is still performed at 3.6 (average number of CGs assigned to each BG) candidate spectra. These likelihood evaluations lead to a computationally expensive implementation. State-based MC MDT yields WERs fairly close to those obtained in other MC MDT experiments, but with a significantly higher computational cost, as shown in the 8th rows in Tables 2, 3, 4, and 5.

6.5. Cluster-based reconstruction

6.5.1. Cluster-based reconstruction with MAP

This experiment is an instance of the concept formulated by Equations (4), (5), and (6). Each of the 500 full-covariance spectral CGs is used to impute clean speech with six iterations of the MAP imputation. The corresponding diagonal-covariance CGs are used to calculate the definite integrals in Equation (6). To compensate for unmodeled correlations, the integrals are exponentiated with 0.3, a value that is tuned on the test set for best accuracy. The global clean spectrum is then reconstructed using Equation (4). Since the likelihoods of the 500 CGs are already calculated, they are used to select BGs as explained in Section 4.4. Despite the test set optimization, the cluster-based reconstruction with MAP imputation is still less robust than the MC MDT systems when comparing row 7 with rows 3, 4, 5, and 8 in Tables 2 and 3. The use of a more accurate speech model provided by the BGs seems to pay off.

6.5.2. Cluster-based reconstruction with GD

This system approximates the previous one by combining the imputed spectra obtained from the PROSPECT CGs like in Equation (4), but takes a different approach to compute the posterior probabilities of the CGs. These posteriors are calculated by renormalizing the likelihoods of the imputed clean spectra. The likelihoods also serve for BG selection as explained in Section 4.4. To compensate for unmodeled correlations, the likelihoods are exponentiated with 0.3, a value that is also tuned on the test set for best accuracy. We did not observe a significant accuracy gain beyond the 500 PROSPECT CGs used to report the results in the tables. The approximations outlined above do not harm the robustness as shown by a comparison between the rows 6 and 7 of Tables 2 and 3, but this implementation is faster because GD imputation is more efficient than MAP and the computation of the posterior probabilities are simplified. Again, despite the test set optimization applied for this cluster-based reconstruction method, MC MDT outperforms it as well.

6.6. Imputation using log-spectral and cepstral CGs

Using a PROSPECT feature representation for the CGs in MC MDT experiments is an implementation choice motivated by speed considerations (see Section 4.1). The CGs can also be trained with other features, e.g., cepstra or log-spectra. To accomplish the comparison of the MC MDT system using CGs in these domains, same number, namely 500 of cepstral CGs and log-spectral CGs are trained using the same data-driven approach as described in Section 6.1.3.

6.6.1. Cepstral imputation

The dimension of the cepstral CGs is 39: 13 static cepstral coefficients, 13 first-and second-order time derivatives. The average number of CGs per BG is 3.6, the same as for PROSPECT CGs. During recognition, the imputation is performed by Multiplicative Updates (MU) [32] with five iterations, an algorithm capable of handling rank-deficient precision matrices, such as $\mathbf{H}_k$

in Equation (11). The testing results are shown in row 10 of Tables 2, 3, 4, and 5. The WER and the percentage of selected BGs obtained by using cepstral CGs are comparable with using PROSPECT CGs. Observe that cepstral CGs introduce time-consuming imputation of MU, which slows down the imputation of CGs by 30-60%.

6.6.2. Spectral imputation

The spectral imputation method described by Equation (2) is tempting for its simplicity. It is worth investigating whether it yields a list of candidate spectra of sufficient quality. The dimension of the log-spectral CGs is 66: 22 static log-spectral coefficients and their first- and second-order time derivatives. The average number of CGs per BG is increased to 5. The test results are shown in row 9 of Tables 2, 3, 4, and 5. While it saves time in CG imputation, the method looses both accuracy and efficiency compared to PROSPECT CG imputation. BG selection is also less effective and more BGs need to be activated to guarantee a reasonable accuracy. Finally, spectral CGs are not able to provide channel estimates (as in Section 6.2.1) as PROSPECT CGs can. This claim is motivated by an experiment (not reported in this article) where the log-spectral CGs provide the candidates of clean speech and trigger the BG selection, while PROSPECT CGs are used for channel estimation, which improved the recognition accuracy by 3.58% relative on test sets 8-14.

7. Conclusions and future work

We have proposed several effective optimizations to a large vocabulary speech recognizer that is based on MDT. The outcome is a recognizer that runs equally fast as the uncompensated system, has identical performance on clean data, has the same robustness as our latest published missing data system [23] and shows competitive performance on the AURORA-4 task.

We first formulated the missing data paradigm such that it can be applied to an acoustic model that requires no compromises on accuracy and uses standard feature representations, i.e., a formulation that covers cepstral as well as LDA-features as they are commonly used in today's speech recognizers. This formulation exploits the most accurate speech model that the recognizer disposes of: the backend HMM. The computational load was significantly reduced by the proposed MC approach to solve the CLSQ problems with sufficient accuracy for practical purposes. Here, candidates are obtained from exact solutions on a smaller set of CG, followed by selection of the most likely candidate. The posterior probabilities of the CG were exploited to construct a Gaussian selection algorithm that saves more computation by excluding Gaussians that are unlikely to make a significant contribution to the state likelihoods. Finally, the CGs were structured hierarchically with a model-based Gaussian clustering algorithm to achieve further speed gains.

The proposed method was compared to cluster-based imputation, an MDT that enhances the feature vector based on a GMM speech model, a technique that is also suitable for large vocabulary tasks. Our experiments reveal that it is beneficial to accuracy to exploit the more accurate backend model instead.

The optimizations show that no modeling compromises are required to apply MDT to large vocabulary recognition and that, on noisy data, any additional computational cost in likelihood calculation is easily recovered by a reduction in the search effort. These benefits make the missing data formalism very suitable to tackle various robustness issues beyond the additive noise effects considered in this article. With a suitable missing data detector, the solutions described in this article open pathways to also efficiently cover reverberated speech and exploit multiple microphones to implement directional hearing.

Appendix: Computational complexity of maximized likelihood per Gaussian

This section reveals how the numbers of multiplication involved in each approach in Table 1 are obtained. The complexity is quantified as the number of multiplications or divisions.

MAP

The iterative method of the MAP algorithm in [24] includes the following steps:

a. Initialize $\mathbf{x}_{t,\ u}$ using the component-wise minimization: $\mathbf{x}_{t,\ u}(i) = \min[\boldsymbol{\mu}_{\ t,\ u}(i), \mathbf{y}_{t,\ u}(i)]$ as in Equation (2).

b. For each i of the unreliable sub vector $\mathbf{x}_{t,\ u}$, calculate the conditional mean

$$\bar{\mathbf{x}}_{t,u,k}(i) = E(\mathbf{x}_{t,u,k}(i)|\mathbf{x}_{t,u,k}(1)\ldots\mathbf{x}_{t,u,k}(i-1),\ \mathbf{x}_{t,u,k}(i+1)\ldots\mathbf{x}_{t,u,k}(D-D_{t,u}), \mathbf{y}_{t,r})$$

where $D_{t,\ u}$ is the number of unreliable components in the spectrum at time t.

And constrain

$$\bar{\mathbf{x}}_{t,u,k}(i) \leftarrow \min\left(\mathbf{y}_{t,u,k}(i), \bar{\mathbf{x}}_{t,u,k}(i)\right)$$

c. Repeat step b for several iterations and calculate the likelihood $p(\mathbf{x}_{t,\ u,\ k}, \mathbf{y}_{t,\ r}|k)$

Raj et al. provide a standard solution for the conditional mean in step b:

$$\bar{\mathbf{x}}_{t,u,k} = \boldsymbol{\mu}_{t,u,k} + \boldsymbol{\Sigma}_{t,u,r,k}\left(\boldsymbol{\Sigma}_{t,r,r,k}\right)^{-1}\left(\mathbf{y}_{t,r} - \boldsymbol{\mu}_{t,r,k}\right)$$

where $\boldsymbol{\mu}_{t,u,k}$ is the mean of unreliable sub-vector. $\boldsymbol{\Sigma}_k$ is the covariance of Gaussian k. $\boldsymbol{\Sigma}_{t,u,r,k}$ contains only the rows with indices corresponding to the unreliable components and columns with the reliable indices in $\boldsymbol{\Sigma}_{t,k}$.

The conditional mean can also be formulated as

$$\bar{\mathbf{x}}_{t,u,k} = \boldsymbol{\mu}_{t,u,k} - (\mathbf{H}_{t,u,u,k})^{-1}\mathbf{H}_{t,u,r,k}(\mathbf{y}_{t,r} - \boldsymbol{\mu}_{t,r,k})$$

where $\mathbf{H}_k = \boldsymbol{\Sigma}_k^{-1}$ and $\mathbf{H}_{t,u,u,k}$ is a $D_{t,u}$ by $D_{t,u}$ sub-matrix of $\mathbf{H}_k$ containing only the rows and columns corresponding to the unreliable components in the feature vector. $\mathbf{H}_{t,u,r,k}$ is a $D_{t,u}$ by D-$D_{t,u}$ sub-matrix of $\mathbf{H}_k$ containing only the rows corresponding to the indices of unreliable components in the feature and the columns with reliable indices.

In step b, only 1 dimension is free and updated per operation as:

$$\bar{\mathbf{x}}_{t,u,k}(i) = \boldsymbol{\mu}_{t,u,k}(i) - \frac{\sum_{j\neq i, x(j)\in\mathbf{x}_{t,u,k}} \mathbf{H}_k(i,j)(\mathbf{x}_t(j) - \boldsymbol{\mu}_k(j)) + \sum_{y(m)\in\mathbf{y}_{t,r}} \mathbf{H}_k(i,m)(\mathbf{y}_t(m) - \boldsymbol{\mu}_k(m))}{\mathbf{H}_k(i,i)}$$

The second part of the summation in the numerator is constant and can be calculated at the first iteration. Hence, the MAP algorithm involves $(D_{t,u} + 1) \times D_{t,u}$ multiplications per iteration where the dimension of $\bar{\mathbf{x}}_{t,u,k}$ is $D_{t,u}$. In the above equation, $\mathbf{x}_t(j)$ is the jth component of the latest updated unreliable component.

Besides updating the clean speech in each iteration, the likelihood of the CG is also required for BG selection, which involves $(D + 1)D$ multiplications. Each iteration of MAP is very efficient, but as shown by the experimental result, MAP needs six iterations to reach convergence in terms of WER. Furthermore, as described in Section 4.1, the full precision matrix has to be handled in the CPU cache memory. Hence, MAP is slower than GD + PROSPECT.

Multiplicative updates

As outlined in [32,33], the step calculation of the ith unreliable component for Gaussian k is given by

$$\mathbf{x}_u(i) \leftarrow \mathbf{x}_u(i)\frac{\mathbf{b}(i) + \sqrt{\mathbf{b}(i)^2 + 4\left[\mathbf{H}_{t,u,u,k}{}^{+}(\mathbf{x}_{t,u,k} - \boldsymbol{\mu}_{t,u,k})\right]_i\left[\mathbf{H}_{t,u,u,k}{}^{-}(\mathbf{x}_{t,u,k} - \boldsymbol{\mu}_{t,u,k})\right]_i}}{2\left[\mathbf{H}_{t,u,u,k}{}^{+}(\mathbf{x} - \boldsymbol{\mu}_{t,u,k})\right]_i}$$

$\mathbf{H}_{t,u,u,k}{}^{+}(\mathbf{H}_{t,u,u,k}{}^{-})$ is obtained from $\mathbf{H}_{t,u,u,k}$ by setting all negative (positive) entries to zero. $\mathbf{H}_{t,u,u,k}{}^{+}(\mathbf{x}_{t,u,k}$-$\boldsymbol{\mu}_{t,u,k})$ together with $\mathbf{H}_{t,u,u,k}{}^{-}(\mathbf{x}_{t,u,k}$-$\boldsymbol{\mu}_{t,u,k})$ involve $D_{t,u} \times D_{t,u}$ multiplication operations.

$\mathbf{b} = \mathbf{H}_{t,u,r,k}(\mathbf{y}_{t,r} - \boldsymbol{\mu}_{r,k}) + \mathbf{H}_{t,u,u,k}(\mathbf{x}_{t,u,k} - \boldsymbol{\mu}_{t,u,k})$ so its first term can be calculated prior to iteration. The second term is already calculated while calculating $\mathbf{H}_{t,u,u,k}{}^{+}(\mathbf{x}_{t,u,k} - \boldsymbol{\mu}_{t,u,k})$ and $\mathbf{H}_{t,u,u,k}{}^{-}(\mathbf{x}_{t,u,k} - \boldsymbol{\mu}_{t,u,k})$. The square, multiplication (2 ×) and division in the above equation involves $4 \times D_{t,u}$ multiplications per iteration. In addition, $D_{t,u}$ computationally expensive square root operations are also involved in each iteration. The calculation of the likelihood involves $(D + 1)$ D multiplications, the same as MAP. MU shares the same drawback with MAP that it has to handle the full precision matrix whenever it is called. As proved by the experiments, MU needs five iterations for convergence of the WER.

Gradient descent + cepstral Gaussians

To calculate the likelihood and gradient of a Cepstral Gaussian with diagonal covariance given a frame of unreliable spectrum, the precision matrix of the Gaussian must be either pre-calculated or calculated on-line. The pre-calculation implies that the system has to handle the $D \times D$ precision matrix which leads to frequent data exchange between CPU and main memory. To calculate the gradient on-line, the precision matrix must be represented in the log-spectral domain as

$$\mathbf{H}_k\left(\begin{bmatrix}\mathbf{y}_{t,r}\\ \mathbf{x}_{t,u}\end{bmatrix} - \boldsymbol{\mu}_k\right) = \mathbf{C}'\boldsymbol{\Sigma}_k^{-1}\mathbf{C}\left(\begin{bmatrix}\mathbf{y}_{t,r}\\ \mathbf{x}_{t,u}\end{bmatrix} - \boldsymbol{\mu}_k\right) \qquad (26)$$

where $\mathbf{H}_k$ is the precision matrix of Gaussian k, and it is transformed from the inverse diagonal covariance matrix $\boldsymbol{\Sigma}_k^{-1}$ by applying the transpose of DCT matrix $\mathbf{C}$. Let $\mathbf{C}_r$ be the D_m by D - $D_{t,u}$ sub matrix of $\mathbf{C}$, containing the columns with the reliable indices, and $\mathbf{C}_u$ contains the remaining elements. Equation (26) can be represented as

$$\begin{aligned}\mathbf{C}'\boldsymbol{\Sigma}_k^{-1}\mathbf{C}\left(\begin{bmatrix}\mathbf{y}_{t,r}\\ \mathbf{x}_{t,u}\end{bmatrix} - \boldsymbol{\mu}_k\right) &= \begin{bmatrix}\mathbf{C}_r'\\ \mathbf{C}_u'\end{bmatrix}\boldsymbol{\Sigma}_k^{-1}\begin{bmatrix}\mathbf{C}_r & \mathbf{C}_u\end{bmatrix}\left(\begin{bmatrix}\mathbf{y}_{t,r}\\ \mathbf{x}_{t,u}\end{bmatrix} - \boldsymbol{\mu}_k\right)\\ &= \mathbf{C}'\boldsymbol{\Sigma}_k^{-1}\mathbf{C}_r\left(\mathbf{y}_{t,r} - \boldsymbol{\mu}_{t,r,k}\right) + \begin{bmatrix}\mathbf{C}_u'\boldsymbol{\Sigma}_k^{-1}\mathbf{C}_u\left(\mathbf{x}_{t,u} - \boldsymbol{\mu}_{t,u,k}\right)\\ \mathbf{C}'_r\boldsymbol{\Sigma}_k^{-1}\mathbf{C}_u\left(\mathbf{x}_{t,u} - \boldsymbol{\mu}_{t,u,k}\right)\end{bmatrix}\end{aligned}$$

$\mathbf{C}'\boldsymbol{\Sigma}_k^{-1}\mathbf{C}_r\left(\mathbf{y}_{t,r} - \boldsymbol{\mu}_{t,r,k}\right)$ is constant for every iteration and is part of the calculation of the likelihood. $\mathbf{C}'\boldsymbol{\Sigma}_k^{-1}\mathbf{C}_u\left(\mathbf{x}_{t,u} - \boldsymbol{\mu}_{t,u,k}\right)$ involves $D_{t,u}(2D_m + 1)$ multiplications for the unreliable component of the gradient. A small fraction of the gradient needs to be added to the gradient to cope with the singularity of $\mathbf{H}_k$ as mentioned in Equation (11). Hence, another $D_{t,u}$ multiplications are needed. The calculation of step size and step scale in Equation (13) involve $D_mD + D_m + D + D_{t,u}$ multiplications. The calculation of likelihood contains $2D_mD + D_m + D$ multiplications.

Gradient descent + PROSPECT features

The cost function for a Gaussian trained with PROSPECT features is shown in Equation (17). The calculation of gradient $\mathbf{g}_{t,k}$ can be decomposed to the projection part and cepstral part. The following quantities must be calculated.

$\mathbf{C}_c\left(\mathbf{x}_t - \mathbf{R}'\boldsymbol{\mu}_k\right)$: $\mathbf{R}'\boldsymbol{\mu}_k$ is the spectral mean of Gaussian k which is pre-calculated from the PROSPECT mean using the inverse PROSPECT transform $\mathbf{R}$. $D_cD_{t,\ u}$ multiplication are involved for the unreliable components every iteration. Additionally $D_c(D - D_{t,\ u})$ multiplications are required to calculate the reliable components before the first iteration.

$\mathbf{C}_c'\mathbf{C}_c\left(\mathbf{x}_t - \mathbf{R}'\boldsymbol{\mu}_k\right)$: Based on the calculation of $\mathbf{C}_c\left(\mathbf{x}_t - \mathbf{R}'\boldsymbol{\mu}_k\right)$, D_cD multiplications are involved.

$\boldsymbol{\Sigma}_k^{\perp^{-1}}\left(\mathbf{I} - \mathbf{C}_c'\mathbf{C}_c\right)\left(\mathbf{x}_t - \mathbf{R}'\boldsymbol{\mu}_k\right)$: includes another D multiplications.

$\mathbf{C}_c\boldsymbol{\Sigma}_k^{\perp^{-1}}\left(\mathbf{I} - \mathbf{C}_c'\mathbf{C}_c\right)\left(\mathbf{x}_t - \mathbf{R}'\boldsymbol{\mu}_k\right)$: includes another D_cD multiplications.

$\mathbf{C}_c'\mathbf{C}_c\boldsymbol{\Sigma}_k^{\perp^{-1}}\left(\mathbf{I} - \mathbf{C}_c'\mathbf{C}_c\right)\left(\mathbf{x}_t - \mathbf{R}'\boldsymbol{\mu}_k\right)$: requires $D_cD_{t,\ u}$ multiplications per iteration.

The cepstral part $\mathbf{C}_c'\boldsymbol{\Sigma}_k^{c^{-1}}\mathbf{C}_c\left(\mathbf{x}_t - \mathbf{R}'\boldsymbol{\mu}_k\right)$: Based on the calculation of $\mathbf{C}_c\left(\mathbf{x}_t - \mathbf{R}'\boldsymbol{\mu}_k\right)$, $D_c + D_cD_{t,\ u}$ multiplications per iteration are involved.

Given the obtained gradient $\mathbf{g}_{t,\ k}$, the step size involves the following quantities as in Equation (13):

$\mathbf{g}_{t,\ k}'\mathbf{g}_{t,\ k}$: D multiplications.

$\mathbf{C}_c\mathbf{g}_{t,\ k}$: D_cD multiplications.

$\mathbf{g}_{t,k}'\mathbf{C}_c'\boldsymbol{\Sigma}_k^{c^{-1}}\mathbf{C}_c\mathbf{g}_{t,k}$: $2D_c$ multiplications.

$(\mathbf{I}\text{-}\mathbf{C}_c'\mathbf{C}_c)\mathbf{g}_{t,\ k}$: D_cD multiplications given $\mathbf{C}_c\mathbf{g}_{t,k}$.

$\mathbf{g}_{t,k}'\left(\mathbf{I} - \mathbf{C}_c'\mathbf{C}_c\right)\boldsymbol{\Sigma}_k^{\perp^{-1}}\left(\mathbf{I} - \mathbf{C}_c'\mathbf{C}_c\right)\mathbf{g}_{t,k}$: $2D$ multiplications

Another D multiplications are required for scaling the gradient as explained in the previous section.

Besides the iterations, the initial likelihood involves 2 $(D + D_c)$ multiplications.

With the typical values of D, D_m, D_c and $D_{t,\ u}$ in Table 1 the number of multiplications involved in MAP is 2138 per Gaussian, while it is 2106 for MU. But MU needs 80 square root operations per Gaussian. The number of multiplications involved in GD with cepstral Gaussian is 2177. This number is reduced to 1416 when using PROSPECT features.

Abbreviations

ASA: auditory scene analysis; BE: BackEnd; BG: Backend Gaussian; CCG: cluster-of-cluster Gaussians; CG: cluster Gaussian; CGS: cluster Gaussian selection; CLBR: cluster-based reconstruction; CLSQ: constrained least squares; DCT: discrete cosine transform; EM: expectation maximization; FRoG: fast removal of Gaussians; GB: Gaussian-based; GD: gradient descent; KLD: Kullback-Leibler divergence; LDA: linear discriminant analysis; MAP: maximum a posterior probability; MC: multi-candidate; MDT: missing data technique; MFCC: MEL Frequency Cepstral Coefficients; MIDA: mutual information-based discriminant analysis; ML: maximum likelihood; MLE: maximum likelihood estimation; MU: multiplicative updates; PDF: probability density functions; PDT: phonetic decision tree; PMC: parallel model combination; PROSPECT: PRojected SPECTra; SB: state-based; SKLD: symmetric Kullback-Leibler divergence; WSKLD: weighted Kullback-Leibler divergence; WSS: wide-sense stationary.

Acknowledgements

This study was financed by the MIDAS project of the Nederlandse Taalunie under the STEVIN programme. Thanks to Kris Demuynck for various implementations and the anonymous reviewers for suggesting additional interesting comparisons and analyses.

Competing interests

The authors declare that they have no competing interests.

Received: 22 August 2011 Accepted: 29 May 2012
Published: 29 May 2012

References

1. ETSI standard doc., Speech processing, transmission and quality aspects (STQ); distributed speech recognition; advanced frontend feature extraction algorithm; compression algorithms. *ETSI, Tech Rep ES 202 050 v1.1.3* (2003)
2. JAN Flores, SJ Young, Continuous speech recognition in noise using spectral subtraction and HMM adaptation, in *Proceedings of ICASSP*, Adelaide, South Australia, Australia, 409–412 (1994)
3. J Droppo, L Deng, A Acero, Evaluation of the SPLICE algorithm on the Aurora2 database, in *Proceedings of Eurospeech*, Aalborg, Denmark, pp. 217–220 (2001)
4. PJ Moreno, B Raj, RMJ Stern, A vector Taylor series approach for environment-independent speech recognition, in *Proceedings of ICASSP*, Atlanta, Georgia, USA, 733–736 (1996)
5. M Gales, *Model-based techniques for noise robust speech recognition*, PhD thesis, (University of Cambridge, September 1995)
6. CJ Leggetter, PC Woodland, Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models. Comput Speech Lang. **9**(2), 171–185 (1995). doi:10.1006/csla.1995.0010
7. JL Gauvain, CH Lee, Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains. IEEE Trans Speech Audio Process. **2**(2), 291–298 (1994). doi:10.1109/89.279278
8. AS Bregman, *Auditory Scene Analysis* (MIT Press, Cambridge, 1990)
9. M Van Segbroeck, H Van hamme, Vector quantization based mask estimation for missing data ASR, in *Proceedings of Interspeech*, Antwerp, Belgium, 910–913 (2007)
10. J Barker, M Cooke, P Green, Robust ASR based on clean speech models: an evaluation of missing data, in *Proceedings of Eurospeech*, Aalborg, Denmark, 213–216 (2001)
11. M Cooke, A glimpsing model of speech perception in noise. J Acoust Soc Am. **119**(3), 1562–1573 (2006). doi:10.1121/1.2166600
12. JP Barker, MP Cooke, DPW Ellis, Decoding speech in the presence of other sources. Speech Commun. **45**(1), 5–25 (2005). doi:10.1016/j.specom.2004.05.002
13. S Srinivasan, D Wang, Transforming binary uncertainties for robust speech recognition. IEEE Trans Audio Speech Lang Process. **15**(7), 2130–2140 (2007)
14. S Srinivasan, D Wang, Robust speech recognition by integrating speech separation and hypothesis testing. Speech Commun. **52**(1), 72–81 (2010). doi:10.1016/j.specom.2009.08.008
15. M Cooke, P Green, L Josifovski, A Vizinho, Robust automatic speech recognition with missing and unreliable acoustic data. Speech Commun. **34**(3), 267–285 (2001). doi:10.1016/S0167-6393(00)00034-0
16. RP Lippmann, BA Carlson, Using missing feature theory to actively select features for robust speech recognition with interruptions, filtering and noise, in *Proceedings of Eurospeech*, Rhodes, Greece, 37–40 (1997)
17. ML Seltzer, B Raj, RM Stern, A Bayesian classifier for spectrographic mask estimation for missing feature speech recognition. Speech Commun. **43**(4), 379–393 (2004). doi:10.1016/j.specom.2004.03.006
18. M Cooke, A Morris, P Green, Missing data techniques for robust speech recognition, in *Proceedings of ICASSP*, Munich, Germany, 863–866 (1997)
19. P Renevey, A Drygajlo, Detection of reliable features for speech recognition in noisy conditions using a statistical criterion, in *Proceedings of CRAC Workshop*, Aalborg, Denmark, 71–74 (2001)
20. S Srinivasan, Y Shao, Z Jin, D Wang, A computational auditory scene analysis system for robust speech recognition, in *Proceedings of Interspeech*, Pittsburg, Pennsylvania, USA, 73–76 (2006)

21. C Cerisara, S Demange, J-P Haton, On noise masking for automatic missing data speech recognition: a survey and discussion. Comput Speech Lang. **21**(3), 443–457 (2007). doi:10.1016/j.csl.2006.08.001
22. A Coy, J Barker, Soft harmonic masks for recognising speech in the presence of a competing speaker, in *Proceedings of Interspeech*, Lisbon, Portugal, 2641–2644 (2005)
23. M Van Segbroeck, H Van hamme, Advances in missing feature techniques for robust large vocabulary continuous speech recognition. IEEE Trans Audio Speech Lang Process. **19**(1), 123–137 (2011)
24. B Raj, ML Seltzer, RM Stern, Reconstruction of missing features for robust speech recognition. Speech Commun. **43**(4), 275–296 (2004). doi:10.1016/j.specom.2004.03.007
25. H Van hamme, Robust speech recognition using cepstral domain missing data techniques and noisy masks, in *Proceedings of ICASSP*, Montreal, Quebec, Canada, **1**, 213–216 (2004)
26. C Cerisara, Towards missing data recognition with cepstral features, in *Proceedings of Eurospeech*, Geneva, Switzerland, 3057–3060 (2003)
27. J Häkkinen, H Haverinen, On the use of missing feature theory with cepstral features, in *Proceedings of CRAC Workshop*, Aalborg, Denmark, (2001)
28. F Faubel, J McDonough, D Klakow, Bounded conditional mean imputation with Gaussian mixture models: a reconstruction approach to partly occluded features, in *Proceedings of ICASSP*, Taipei, Taiwan, 3869–3872 (2009)
29. R Haeb-Umbach, H Ney, Linear discriminant analysis for improved large vocabulary continuous speech recognition, in *Proceedings of ICASSP*, San Francisco, California, USA, 13–16 (1992)
30. N Kumar, AG Andreou, A generalization of linear discriminant analysis in maximum likelihood framework, *Tech Rep JHU-CLSP Technical Report No. 16*, (Johns Hopkins University, August 1996)
31. R Fletcher, *Practical Methods of Optimization* (John Wiley & Sons, Chichester, 1980)
32. LK Saul, F Sha, DD Lee, Statistical signal processing with non-negativity constraints, in *Proceedings of Eurospeech*, Geneva, Switzerland, 1001–1004 (2003)
33. H Van hamme, Prospect features and their application to missing data techniques for robust speech recognition, in *Proceedings of Interspeech*, Jeju, Korea, 101–104 (2004)
34. J Duchateau, K Demuynck, D Van Compernolle, P Wambacq, Class definition in discriminant feature analysis, in *Proceeding of Eurospeech*, Aalborg, Denmark, 1621–1624 (2001)
35. SJ Young, JJ Odell, PC Woodland, Tree-based state tying for high accuracy acoustic modelling, in *Proceedings of Workshop on Human Language Technology*, Plainsboro, New Jerey, USA, 307–312 (1994)
36. D Iskra, B Grosskopf, K Marasek, H van den Heuvel, F Diehl, A Kiessling, SPEECON–Speech Databases for Consumer Devices: Database Specification and Validation, in *Proceedings of LREC*, Las Palmas, Spain, pp. 329–333 (2002)
37. K Demuynck, *Extracting, modeling and combining information in speech recognition*, PhD thesis, K.U., Leuven, ESAT, (2001)
38. J Fritsch, I Rogina, The bucket box intersection (BBI) algorithm for fast approximate evaluation of diagonal mixture Gaussians, in *Proceeding of ICASSP*, Atlanta, Georgia, USA, **2**, 273–276 (1996)
39. E Bocchieri, Vector quantization for efficient computation of continuous density likelihoods, in *Proceeding of ICASSP*, Minneapolis, Minnesota, USA, **2**, 692–695 (1993)
40. T Watanabe, K Shinoda, K Takagi, KI Iso, High speed speech recognition using tree-structured probability density function, in *Proceeding of ICASSP*, Detroit, Michigan, USA, **1**, 556–559 (1995)
41. Y Wang, H Van hamme, Speed improvements in a missing data-based speech recognizer by Gaussian selection, in *Proceedings of NAG-DAGA*, Rotterdam, Netherlands, 423–426 (2009)
42. DJ Mackay, *Information Theory, Inference, and Learning Algorithms* (Cambridge University Press, Cambridge, UK, 2003)
43. TA Myrvoll, FK Soong, Optimal clustering of multivariate normal distributions using divergence and its application to HMM adaptation, in *Proceedings of ICASSP*, Hong Kong, pp. 552–555 (2003)
44. N Parihar, J Picone, Analysis of the aurora large vocabulary evaluations, in *Proceedings of Eurospeech*, Geneva, Switzerland, 337–340 (2003)
45. JF Gemmeke, M Van Segbroeck, Y Wang, B Cranen, H Van hamme, Automatic speech recognition using missing data techniques: handling of real-world data, in *Robust Speech Recognition of Uncertain or Missing Data*, ed. by R, Haeb-Umbach, D Kolossa Berlin-Heidelberg (Germany), Springer Verlag, pp. 157–186 (2011)
46. S Young, D Kershaw, J Odell, D Ollason, V Valtchev, P Woodland, *The HTK Book* http://htk.eng.cam.ac.uk/docs/docs.shtml
47. SPRAAK, Speech Processing, Recognition and Automatic Annotation Kit http://www.spraak.org/

doi:10.1186/1687-4722-2012-17
Cite this article as: Wang and Van hamme: **Multi-candidate missing data imputation for robust speech recognition.** *EURASIP Journal on Audio, Speech, and Music Processing* 2012 **2012**:17.

Submit your manuscript to a SpringerOpen journal and benefit from:

- ► Convenient online submission
- ► Rigorous peer review
- ► Immediate publication on acceptance
- ► Open access: articles freely available online
- ► High visibility within the field
- ► Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com