Importance Sampling to Compute Likelihoods of Noise-Corrupted Speech[☆]

R. C. van Dalen[*], M. J. F. Gales

Engineering Department, University of Cambridge, Trumpington Street, Cambridge CB2 1PZ, United Kingdom

Abstract

One way of making speech recognisers more robust to noise is *model compensation*. Rather than enhancing the incoming observations, model compensation techniques modify a recogniser's state-conditional distributions so they model the speech in the target environment. Because the interaction between speech and noise is non-linear, even for Gaussian speech and noise the corrupted speech distribution has no closed form. Thus, model compensation methods approximate it with a parametric distribution, such as a Gaussian or a mixture of Gaussians. The impact of this approximation has never been quantified. This paper therefore introduces a non-parametric method to compute the likelihood of a corrupted speech observation. It uses sampling and, given speech and noise distributions and a mismatch function, is exact in the limit. It therefore gives a theoretical bound for model compensation. Though computing the likelihood is computationally expensive, the novel method enables a performance comparison based on the criterion that model compensation methods aim to minimise: the KL divergence to the ideal compensation. It gives the point where the Kullback-Leibler (KL) divergence is zero. This paper examines the performance of various compensation methods, such as vector Taylor series (VTS) and data-driven parallel model combination (DPMC). It shows that more accurate modelling than Gaussian-for-Gaussian compensation improves the performance of speech recognition.

Keywords: Speech recognition, noise-robustness

1. Introduction

Changes in background noise, resulting in a mismatch between training and testing conditions, can severely impact the performance of speech recognition systems. There are two categories of approaches for dealing with this problem. *Feature enhancement* finds a reconstruction of the clean speech, which is then used in decoding. Alternatively, the speech recogniser can be modified to model the distribution of the corrupted speech directly. This is called *model compensation*. However, since the interaction between speech and noise is non-linear, the corrupted-speech distribution has no closed form. In particular, even if the speech and noise are Gaussian-distributed, the corrupted speech is not. However, the majority of schemes (Gales, 1995; Sagayama et al., 1997; Acero et al., 2000; Li et al., 2010; Seltzer et al., 2010; van Dalen and Gales, 2011) assume the corrupted speech to be Gaussian-distributed. Thus, improvements from different ways of computing the same form of distribution are limited (see e.g. Li et al., 2010). There has been research into non-Gaussian schemes (Kristjansson and Frey, 2002; Myrvoll and Nakamura, 2004), which are slow and impractical, and they do not match the exact corrupted speech distribution. It is therefore unknown how great the impact of the Gaussian assumption is.

Instead of assuming a specific parametric form, this work will introduce a non-parametric method to compute the corrupted speech likelihood. It will assume that distributions of the speech and noise and their interaction are known. The central intuition is that the corrupted speech density only needs to be evaluated at points given by the observed feature vector. The exact likelihood for one observation vector is an integral over the clean speech and the noise.

[☆]This work was part-funded by Toshiba Research Europe Ltd., Cambridge Research Laboratory.

[*]Corresponding author. Tel: (+44) 1223 765152

Email addresses: rcv25@cam.ac.uk (R. C. van Dalen), mjfg@eng.cam.ac.uk (M. J. F. Gales)

This integral will be approximated dimension by dimension with sequential importance re-sampling. In the limit, this yields the exact likelihood given models for the speech and the noise, and a mismatch function.

Though too expensive for speech recognition, the approach can be used to evaluate the accuracy of model compensation or enhancement schemes. It is possible to assess the closeness to the real distribution for speech recogniser compensation methods, with a metric based on the KL divergence. In the limit, the new sampling method will effectively give the point where the KL divergence is zero, which is otherwise not known. This calibration will make it possible to determine how far well-known compensation methods are from the optimal distribution.

This work will also examine how well the KL divergence predicts speech recogniser word error rate. It will compare different compensation schemes, and examine the effect of common approximations. This includes assuming the corrupted speech distribution Gaussian, and approximations to the function that relates speech, noise, and observations.

At the same time as the method to compute corrupted-speech likelihoods in this paper was first published (van Dalen and Gales, 2010), a similar method was proposed (Hershey et al., 2010). This method has three main differences. First, the model is different: no mel-bins are used, so that the model of the *phase factor* (see section 2.1) is different. Also, the variable transformation is different. The most important difference, however, is that the method works dimension-per-dimension. It therefore fails to take correlations in speech and noise Gaussians into account. Section 4.2 will introduce a sampling method for multiple dimensions. The strategy it uses may also apply to the model in Hershey et al. (2010). Another paper about a Monte Carlo method for noise-robustness is Faubel and Wölfel (2007). However, it is otherwise unrelated to this paper: it applies feature enhancement, i.e. it reconstructs clean speech vectors, and applies importance sampling to a different problem than this paper will.

This paper is structured as follows. Section 2 will discuss the corrupted-speech distribution, and in particular properties of the *phase factor*. Section 3 will show how existing methods for model compensation approximate the corrupted speech with parametric distributions. A sampling method to approximate the corrupted-speech likelihood for one observation will be introduced in section 4. Section 5 will introduce a method to quantify how far a model compensation is from optimal, based on the KL divergence. The experiments in section 6 will relate relative performance of compensation methods on the KL divergence to word error rates, to quantify how far performance of compensation methods is from optimal.

2. The noise-corrupted speech

Methods for model compensation (and model-based feature enhancement) usually represent the signals of the noise and corrupted speech in the same domain as the clean speech: the cepstral domain. Cepstral features are related to log-spectral features by the discrete cosine transform (DCT), which is a linear transformation. The reason the cepstral domain is often preferred is that the DCT helps to decorrelate the features within a feature vector. However, correlations do not fully disappear, and especially under noisy conditions it becomes important to model them correctly (Gales and van Dalen, 2007). For the purpose of modelling the interaction between speech, noise, and observations, the log-spectral domain has an advantage: the interaction is per dimension. Therefore, though the recognition experiments will use cepstral-domain models, the theory will use the log-spectral domain.

The model for the clean speech normally uses Gaussians. Canonically, these are the components of a state-conditional mixture. They can also represent a base class (Liao and Gales, 2005), or a mixture component of a simplified mixture of Gaussians for feature enhancement (Ephraim, 1990; Stouten, 2006). The additive noise is usually assumed to be Gaussian distributed. The convolutional (channel) noise is normally assumed constant for one utterance. Since in the log-spectral domain it appears as a constant offset to the speech signal, here it will be assumed zero and omitted from the notation. This work will therefore focus on the corrupted-speech distribution resulting from combining one Gaussian for clean speech $\mathbf{x}$ and one Gaussian for noise $\mathbf{n}$,

$$\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}_{\mathsf{x}}, \boldsymbol{\Sigma}_{\mathsf{x}})\,; \qquad\qquad \mathbf{n} \sim \mathcal{N}(\boldsymbol{\mu}_{\mathsf{n}}, \boldsymbol{\Sigma}_{\mathsf{n}})\,, \tag{1}$$

with known log-spectral domain parameters, and covariances $\boldsymbol{\Sigma}_{\mathsf{x}}$ and $\boldsymbol{\Sigma}_{\mathsf{n}}$ full.

Section 2.1 will discuss the mismatch function, the relation between speech, noise, and corrupted speech used in this work. Section 2.2 will present the resulting corrupted speech distribution, which has no closed form.

2.1. The mismatch function

The relationship between the corrupted speech, the clean speech and the noise is central to noise-robust speech recognition. The term *mismatch function* is often used for the function that takes the speech and noise signals and returns the corrupted speech signal. It is useful to express the mismatch in the log-spectral domain, so that the distribution of the clean speech can inform inference. In the conversion to log-spectral coefficients, phase information is discarded. However, when the speech and noise combine, their phase difference causes random variation in the log-spectral representation of the corrupted speech. This effect is usually assumed absent, but this work will follow Deng et al. (2004) and Leutnant and Haeb-Umbach (2009a,b) and include a *phase factor* in the final mismatch function. Since this is a non-standard aspect, the following derivation will focus on the phase factor.

Background noise is additive in the spectral domain, so that the corrupted speech $Y[k]$, the clean speech $X[k]$, the additive noise $N[k]$ are related with $Y[k] = X[k] + N[k]$ (Acero, 1990). To remove the phase information from the complex spectral coefficients, the magnitude or power spectrum is taken. This paper will assume the power spectrum, in which the mismatch function becomes

$$|Y[k]|^2 = |X[k] + N[k]|^2 = |X[k]|^2 + |N[k]|^2 + 2|X[k]N[k]| \cos\theta_k, \tag{2}$$

where θ_k is the angle in the complex plane between $X[k]$ and $N[k]$. It indicates the phase difference at frequency k between the clean speech and the noise.

The next step is to reduce the number of coefficients, by applying I filter bins to the power-spectral coefficients. Let w_{ik} specify the contribution of the kth frequency to the ith bin. The mel-filtered power spectrum is then given by coefficients $\bar{X}_i, \bar{N}_i, \bar{Y}_i$:

$$\bar{X}_i = \sum_k w_{ik}|X[k]|^2\,; \quad \bar{N}_i = \sum_k w_{ik}|N[k]|^2\,; \quad \bar{Y}_i = \sum_k w_{ik}|Y[k]|^2 = \bar{X}_i + \bar{N}_i + \sum_k w_{ik}(2|X[k]||N[k]|\cos\theta_k)\,. \tag{3a}$$

This expression for $\bar{Y}_i$ still depends on individual spectral coefficients $X[k], N[k]$. To remove this dependency, a variable α_i, the *phase factor* is introduced:

$$\bar{Y}_i = \sum_k w_{ik}|Y[k]|^2 = \bar{X}_i + \bar{N}_i + 2\alpha_i\sqrt{\bar{X}_i\bar{N}_i}, \qquad \alpha_i \triangleq \frac{\sum_k w_{ik}|X[k]||N[k]|\cos\theta_k}{\sqrt{\bar{X}_i\bar{N}_i}}. \tag{3b}$$

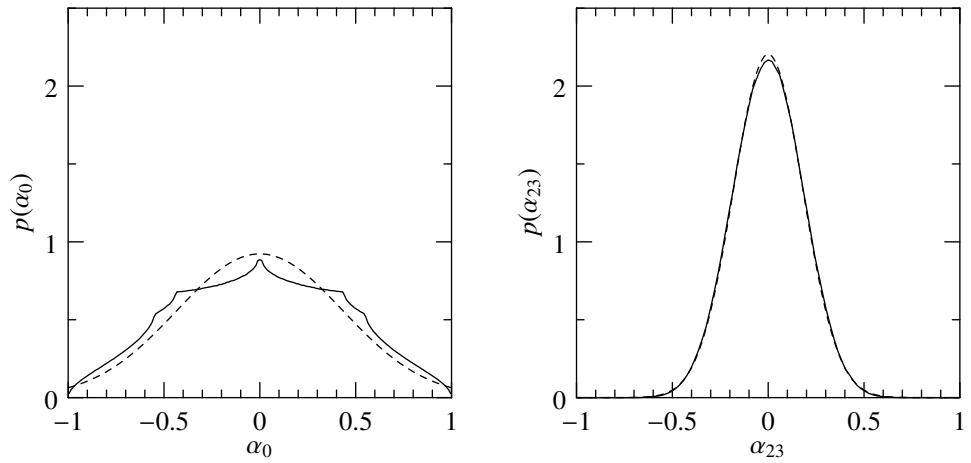
α_i encapsulates the phase difference between the two signals (speech and noise) that are added in one mel-bin. The phase information is discarded in the conversion to the log-spectral domain, so that with speech and noise models in that domain, the phase factor α_i is a random variable. The next subsection will find that under mild assumptions, α_i can be approximated with a truncated Gaussian distribution.

The mel-power-spectral coefficients are usually converted to their logarithms so that $y_i = \log(\bar{Y}_i)$, $x_i = \log(\bar{X}_i)$, and $n_i = \log(\bar{N}_i)$. The mismatch expression in (3a) then can be written in the log-spectral domain, per coefficient or in vector notation:

$$\exp(y_i) = \exp(x_i) + \exp(n_i) + 2\alpha_i \exp\left(\tfrac{1}{2}(x_i + n_i)\right); \qquad \mathbf{exp}(\mathbf{y}) = \mathbf{exp}(\mathbf{x}) + \mathbf{exp}(\mathbf{n}) + 2\boldsymbol{\alpha} \circ \mathbf{exp}\Big(\tfrac{1}{2}(\mathbf{x} + \mathbf{n})\Big), \tag{4}$$

where the corrupted speech vector $\mathbf{y}$ consists of elements y_i, and similarly for $\mathbf{x}$, $\mathbf{n}$, and $\boldsymbol{\alpha}$, and $\mathbf{exp}(\cdot)$ and $\circ$ denote element-wise exponentiation and multiplication.

Before discussing a state-of-the-art approach that aims to approximate the distribution of $\boldsymbol{\alpha}$ closely, it is worth noting a different approach. In practice, the mismatch function is often approximated by leaving out the phase factor term (Gales, 1995; Moreno, 1996; Acero et al., 2000). To make up for this approximation, and potentially for other approximations, it is possible to introduce parameters that can be optimised for a specific task. The reasoning is as follows. Model compensation for noise robustness is a form of adaptation to the data. The parameters that in theory make up the noise model are usually estimated from the data, with the aim to maximise the likelihood of the adapted model. In that case, the difference between traditional adaptation with linear transformations and methods for model compensation is the space to which the adapted model are constrained. In both cases, it can be argued that the best choice for this space is the one that yields the lowest word error rate, which does not necessarily reflect the real environment. The mismatch function is one element that determines this space.



(a) Bin 0 is the narrowest, so that α_0 has the least Gaussian-like distribution.

(b) Bin 23 is the widest, so that α_{23} has the most Gaussian-like distribution.

Figure 1: The distribution of α_i for different mel-filter channels i (—), and their Gaussian approximations (- -).

There are various parameterisations of the mismatch function. One is the assumed power of the spectrum (e.g. the magnitude or power spectrum) (Gales, 1995). Another is obtained by setting $\boldsymbol{\alpha}$ to a fixed value (Li et al., 2009). For different corpora, different parameter settings in the mismatch function produce the best word error rate (Liao, 2007; Li et al., 2009; Gales and Flego, 2010). This illustrates that optimising the space of the adapted model by adjusting parameters in the mismatch function can have a positive impact on word error rate. It can make up for approximations in the compensation process (section 6.2 will examine an example of this) or for effects of maximum likelihood for the noise model.

However, here, the interest is in modelling the real environment closely. The next section will therefore discuss the distribution of the phase factor.

The phase factor

Since there is no process in speech production that synchronises the speech phase with the noise phase at a specific frequency, a priori the phase θ_k is uniformly distributed. Similar to the speech and noise, it is assumed independent per frame. Also, the phase is assumed independently distributed for different frequencies k. The distribution of α_i is a weighted average of $\cos\theta_k$ for all frequencies k in the bin (a graph of the distribution for a single bin can be found in Leutnant and Haeb-Umbach, 2009b). α_i can be shown to be in $[-1, +1]$. As the number of frequencies goes up, by the central limit theorem the distribution of α_i becomes closer to a Gaussian (Deng et al., 2004). For lower-frequency bins, the number of frequencies that is summed over is smaller, so the distribution of α_i is expected to be further away from a Gaussian. This effect can be seen in figure 1, which shows the distributions for two values of i. The dashed lines show the Gaussian approximations. For α_0, the Gaussian is least accurate, but still a reasonable approximation. In this work, the distribution of α_i will therefore be assumed Gaussian for each bin i.

Leutnant and Haeb-Umbach (2009a) find an analytic expression for the variance of α_i. The main assumption is that that all spectral coefficients in one filter bin are equal, removing the influence of particular values of the speech and noise signals. By dividing both numerator and denominator in (3b) by $|X[k]||N[k]|$, which is assumed constant with respect to k,

$$\alpha_i \triangleq \frac{\sum_k w_{ik}|X[k]||N[k]|\cos\theta_k}{\sqrt{\left(\sum_k w_{ik}|X[k]|^2\right)\left(\sum_k w_{ik}|N[k]|^2\right)}} \simeq \frac{\sum_k w_{ik}\cos\theta_k}{\sum_k w_{ik}}. \tag{5}$$

The covariance of the Gaussian can be set to the second moment of the real distribution. It can be shown that,

again assuming that all spectral coefficients in one filter bin are equal, (Leutnant and Haeb-Umbach, 2009a)

$$\sigma_{\alpha,i}^2 \triangleq \mathcal{E}\{\alpha_i^2\} = \frac{\sum_k w_{ik}^2}{2\left(\sum_k w_{ik}\right)^2}. \tag{6}$$

This gives values very close to the actual variance of α_i on various subsets of AURORA 2 (Leutnant and Haeb-Umbach, 2009a). This work will therefore approximate the distribution of α_i as a truncated Gaussian with

$$p(\alpha_i) \propto \begin{cases} \mathcal{N}\left(\alpha_i;\, 0,\, \sigma_{\alpha,i}^2\right) & \alpha_i \in [-1, +1]; \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

Evaluating the density at any point requires the normalisation constant $1/\int_{-1}^{+1} \mathcal{N}(\alpha;\, 0,\, \sigma^2)\, d\alpha$, which could be approximated with an approximation to the Gaussian's cumulative distribution function. However, it is straightforward to draw samples from this distribution, by sampling from the Gaussian and rejecting any samples not in $[-1, +1]$.

2.2. Corrupted speech distribution

With the mismatch function $\mathbf{f}(\cdot)$ in (4), the observation vector is known if the vectors for the speech, noise, and the phase factor are known. If the distributions of the three inputs, $p(\mathbf{x})$, $p(\mathbf{n})$, and $p(\boldsymbol{\alpha})$, are known, then the observation distribution can be trivially described using a Dirac delta at the point given by the mismatch function $\mathbf{f}(\cdot)$:

$$p(\mathbf{y}) = \int p(\mathbf{x}) \int p(\mathbf{n})\, p(\mathbf{y}|\mathbf{x}, \mathbf{n})\, d\mathbf{n}\; d\mathbf{x} \tag{8a}$$

$$= \int p(\mathbf{x}) \int p(\mathbf{n}) \int p(\boldsymbol{\alpha})\, p(\mathbf{y}|\mathbf{x}, \mathbf{n}, \boldsymbol{\alpha})\, d\boldsymbol{\alpha}\; d\mathbf{n}\; d\mathbf{x} \tag{8b}$$

$$= \int p(\mathbf{x}) \int p(\mathbf{n}) \int p(\boldsymbol{\alpha})\, \delta_{\mathbf{f}(\mathbf{x},\mathbf{n},\boldsymbol{\alpha})}(\mathbf{y})\, d\boldsymbol{\alpha}\; d\mathbf{n}\; d\mathbf{x}. \tag{8c}$$

This expression is exact given the models for the speech, noise, and the mismatch function. However, since the mismatch function $\mathbf{f}$ is non-linear, there is no closed-form expression for the distribution of $\mathbf{y}$. Figure 2 shows that the distribution can be bimodal. This work will introduce transformation to the integral in (8c) that enable approximating it with a Monte Carlo method. Standard model compensation methods, however, approximate this distribution with a parametric representation, for example a Gaussian (Gales, 1995; Sagayama et al., 1997; Acero et al., 2000; Li et al., 2010; Seltzer et al., 2010; van Dalen and Gales, 2011). Each clean speech Gaussian can then be replaced by the corrupted speech Gaussian, found using the parameters of the original Gaussian as statistics.

Sampling from the corrupted speech distribution

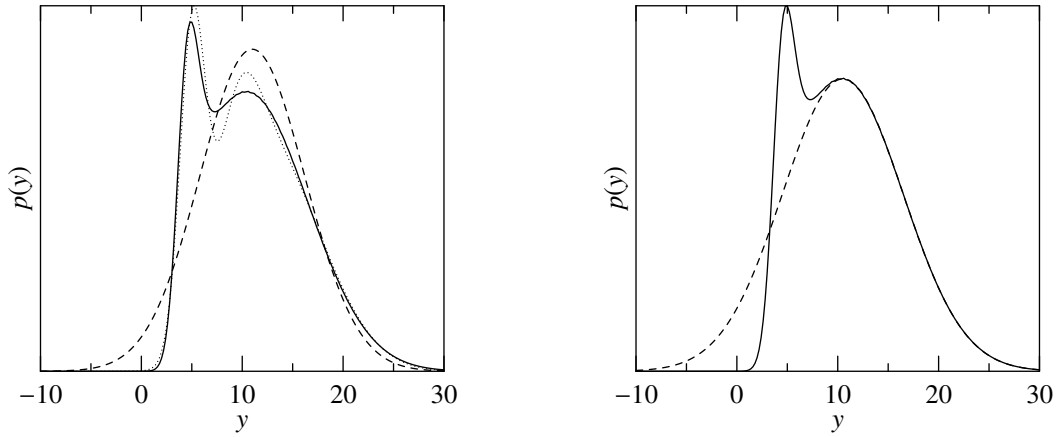
Expressing the corrupted speech distribution parametrically is normally not possible. However, it is straightforward to draw samples $\mathbf{y}^{(l)}$ from the distribution if it is possible to draw samples from the distributions for $\mathbf{x}$, $\mathbf{n}$, and $\boldsymbol{\alpha}$. In this work, the speech will be Gaussian or a mixture of Gaussians; the noise Gaussian. The samples are

$$\mathbf{x}^{(l)} \sim p(\mathbf{x})\,; \qquad \mathbf{n}^{(l)} \sim p(\mathbf{n})\,; \qquad \boldsymbol{\alpha}^{(l)} \sim p(\boldsymbol{\alpha})\,. \tag{9a}$$

The corrupted speech samples are then given by the mismatch function applied to the samples of $\mathbf{x}, \mathbf{n}, \boldsymbol{\alpha}$:

$$\mathbf{y}^{(l)} = \mathbf{f}(\mathbf{x}^{(l)}, \mathbf{n}^{(l)}, \boldsymbol{\alpha}^{(l)}). \tag{9b}$$

Sampling from the corrupted speech distribution will be used in this work to train parametric distributions (DPMC and IDPMC) in section 3.1, and to examine how well approximated distributions match the actual distribution in section 5.



(a) DPMC (– –, Gaussian); iterative DPMC ($\cdots$, GMM). (b) VTS (– –, Gaussian).

Figure 2: The corrupted speech distribution (—) and approximations. $x \sim \mathcal{N}(10.5, 36)$ and $n \sim \mathcal{N}(4, 1)$.

3. Parametric representations for the corrupted speech distribution

There exist several parametric methods for approximating the corrupted speech distribution. Finding a parametric form is normally required for model compensation, which replaces state-conditional clean speech distributions by estimated corrupted speech ("compensated") distributions. This section will discuss two schemes that map one clean speech Gaussian to one corrupted speech Gaussian and one that maps a mixture of Gaussians to another mixture. DPMC compensation (section 3.1) is a sampling approach that only makes the Gaussian approximation at the last instance, by training the Gaussian on samples. Iterative DPMC trains a mixture of Gaussians on samples. VTS compensation (section 3.2) is a standard method that applies a first-order vector Taylor series approximation to the mismatch function, so that the corrupted speech becomes Gaussian. These methods are all carried out offline, before any observations have been seen. The compensation of the Algonquin algorithm (section 3.3), on the other hand, depends on the observation and must be carried out online.

3.1. Data-driven parallel model combination

Data-driven parallel model combination (DPMC) (Gales, 1995) approximates the distributions with samples and applies the exact mismatch function. The Gaussian assumption is made only when training the parameters on the samples. In the limit, it finds the the optimal Gaussian distribution, i.e. the one that yields the highest expected likelihood for the corrupted speech.

The original algorithm did not use phase factor $\boldsymbol{\alpha}$; however, the generalisation is straightforward. Section 2.2 has discussed how to draw samples $\mathbf{y}^{(l)}$ from the corrupted speech distribution. DPMC finds parameters $\boldsymbol{\lambda}$ of a parametric distribution q for the corrupted speech that maximise the likelihood of the samples:

$$\boldsymbol{\lambda} := \arg\max_{\boldsymbol{\lambda}} \sum_{l} \log q(\mathbf{y}^{(l)}; \boldsymbol{\lambda}). \tag{10}$$

This can be interpreted as minimising the KL divergence to the empirical distribution constructed from the samples (van Dalen and Gales, 2011). Standard DPMC finds the maximum-likelihood Gaussian distribution with parameters $\boldsymbol{\lambda} = \{\boldsymbol{\mu}_\mathsf{y}, \boldsymbol{\Sigma}_\mathsf{y}\}$ for the corrupted speech:

$$q(\mathbf{y}; \boldsymbol{\lambda}) = \mathcal{N}(\mathbf{y};\, \boldsymbol{\mu}_\mathsf{y},\, \boldsymbol{\Sigma}_\mathsf{y}); \qquad \boldsymbol{\mu}_\mathsf{y} := \frac{1}{L}\sum_{l=1}^{L} \mathbf{y}^{(l)}; \qquad \boldsymbol{\Sigma}_\mathsf{y} := \frac{1}{L}\sum_{l=1}^{L} \left(\mathbf{y}^{(l)}\mathbf{y}^{(l)\mathsf{T}}\right) - \boldsymbol{\mu}_\mathsf{y}\boldsymbol{\mu}_\mathsf{y}^\mathsf{T}. \tag{11}$$

In the limit as the number of samples goes to infinity, DPMC yields the optimal Gaussian parameters given a mismatch function and distributions for the speech, noise, and phase factor. However, as a large number of samples

are necessary to robustly train the noise-corrupted speech distributions, it is computationally expensive. Also, the real distribution is not Gaussian.

Figure 2 has shown an example of the corrupted speech distribution in one dimension. Even for the one-dimensional case, it is possible to get a bimodal distribution that is impossible to model with one Gaussian. Iterative DPMC (IDPMC) (Gales, 1995) also finds a parametrised distribution for the corrupted speech, but the distribution is a mixture of Gaussians rather than a single Gaussian. This allows it to model the multi-modal nature of the corrupted speech distribution. The word "iterative" in the name of the scheme refers to the iterations of expectation–maximisation necessary to train a mixture of Gaussians. The approximation can become more accurate when the number of Gaussians increases: in the limit as the number of Gaussians M goes to infinity, the mixture of Gaussians becomes equal to the real distribution. However, this requires each component to be trained well, for which it needs sufficient samples. When increasing the number of components M, the number of total samples L must increase by at least the same factor. An iteration of expectation–maximisation takes $O(ML)$ time, so in effect this is at least $O(M^2)$. Additionally, the number of iterations of mixing up, which applies a number of iterations of expectation–maximisation at each step, increases linearly with M. In practice, then, the time complexity of IDPMC is at least $O(M^3)$. This becomes impractical very quickly, especially since modelling the distribution well in a high-dimensional space may require many components.

3.2. Vector Taylor series approximation

Vector Taylor series (VTS) compensation (Moreno, 1996; Acero et al., 2000; Deng et al., 2004) is a standard method for finding Gaussian compensation that is faster than DPMC, though it does not find the optimal Gaussian. Figure 2b shows an example of Gaussian VTS compensation. Rather than approximating the noise-corrupted speech distribution directly, it applies a per-component vector Taylor series approximation to the mismatch function $\mathbf{f}$ in (4). The most important result of this is that, given Gaussians for the clean speech, the noise, and the phase factor, the predicted noise-corrupted speech also becomes Gaussian.

The mismatch function $\mathbf{f}$ can be approximated with a first-order vector Taylor series with expansion at $(\mathbf{x}_0, \mathbf{n}_0, \boldsymbol{\alpha}_0)$ as

$$\mathbf{f}_{\mathsf{vts}}(\mathbf{x}, \mathbf{n}, \boldsymbol{\alpha}) = \mathbf{f}(\mathbf{x}_0, \mathbf{n}_0, \boldsymbol{\alpha}_0) + \mathbf{J}_{\mathsf{x}}(\mathbf{x} - \mathbf{x}_0) + \mathbf{J}_{\mathsf{n}}(\mathbf{n} - \mathbf{n}_0) + \mathbf{J}_{\alpha}(\boldsymbol{\alpha} - \boldsymbol{\alpha}_0), \tag{12a}$$

where the Jacobians for the clean speech, additive noise, and phase factor are

$$\mathbf{J}_{\mathsf{x}} = \left.\frac{\partial \mathbf{y}}{\partial \mathbf{x}}\right|_{\mathbf{x}_0, \mathbf{n}_0, \boldsymbol{\alpha}_0}; \qquad \mathbf{J}_{\mathsf{n}} = \left.\frac{\partial \mathbf{y}}{\partial \mathbf{n}}\right|_{\mathbf{x}_0, \mathbf{n}_0, \boldsymbol{\alpha}_0}; \qquad \mathbf{J}_{\alpha} = \left.\frac{\partial \mathbf{y}}{\partial \boldsymbol{\alpha}}\right|_{\mathbf{x}_0, \mathbf{n}_0, \boldsymbol{\alpha}_0}. \tag{12b}$$

$\boldsymbol{\alpha}$ is approximately Gaussian distributed but constrained to $[-1, +1]$ (see section 2.1). To simplify the distribution of $\mathbf{y}$, this constraint can be ignored, so that $\boldsymbol{\alpha} \sim \mathcal{N}(\boldsymbol{\mu}_{\alpha}, \boldsymbol{\Sigma}_{\alpha})$ with $\boldsymbol{\mu}_{\alpha} = \mathbf{0}$. $\mathbf{f}_{\mathsf{vts}}$ in (12a) then is a sum of linearly transformed independently Gaussian distributed variables, so that the resulting distribution is also Gaussian.

The linearised mismatch function $\mathbf{f}_{\mathsf{vts}}$ replaces $\mathbf{f}$ in the delta function in (8c). The approximation for $\mathbf{y}$ then is the sum of the mismatch function at the expansion point and the three Gaussians:

$$q(\mathbf{y}) := \int \mathcal{N}(\mathbf{x};\, \boldsymbol{\mu}_{\mathsf{x}},\, \boldsymbol{\Sigma}_{\mathsf{x}}) \int \mathcal{N}(\mathbf{n};\, \boldsymbol{\mu}_{\mathsf{n}},\, \boldsymbol{\Sigma}_{\mathsf{n}}) \int \mathcal{N}(\boldsymbol{\alpha};\, \mathbf{0},\, \boldsymbol{\Sigma}_{\alpha})\, \delta_{\mathbf{f}_{\mathsf{vts}}(\mathbf{x},\mathbf{n},\boldsymbol{\alpha})}(\mathbf{y})\, d\boldsymbol{\alpha}\, d\mathbf{n}\, d\mathbf{x} = \mathcal{N}(\mathbf{y};\, \boldsymbol{\mu}_{\mathsf{y}},\, \boldsymbol{\Sigma}_{\mathsf{y}}), \tag{13a}$$

with

$$\boldsymbol{\mu}_{\mathsf{y}} := \mathbf{f}(\mathbf{x}_0, \mathbf{n}_0, \boldsymbol{\alpha}_0) + \mathbf{J}_{\mathsf{x}}(\boldsymbol{\mu}_{\mathsf{x}} - \mathbf{x}_0) + \mathbf{J}_{\mathsf{n}}(\boldsymbol{\mu}_{\mathsf{n}} - \mathbf{n}_0) + \mathbf{J}_{\alpha}(\boldsymbol{\mu}_{\alpha} - \boldsymbol{\alpha}_0); \tag{14a}$$

$$\boldsymbol{\Sigma}_{\mathsf{y}} := \mathbf{J}_{\mathsf{x}}\boldsymbol{\Sigma}_{\mathsf{x}}\mathbf{J}_{\mathsf{x}}^{\mathsf{T}} + \mathbf{J}_{\mathsf{n}}\boldsymbol{\Sigma}_{\mathsf{n}}\mathbf{J}_{\mathsf{n}}^{\mathsf{T}} + \mathbf{J}_{\alpha}\boldsymbol{\Sigma}_{\alpha}\mathbf{J}_{\alpha}^{\mathsf{T}}, \tag{14b}$$

As in the mel-cepstral domain the Jacobians are non-diagonal, the covariance matrices $\boldsymbol{\Sigma}_{\mathsf{y}}$ and $\boldsymbol{\Sigma}_{\mathsf{y}}^{\Delta}$ are full even when the covariances of the clean speech and the noise are assumed diagonal. The expansion points are usually set to the means of the distributions for the clean speech, additive noise and phase factor, so that the terms with the Jacobians in (14a) vanish and the mean becomes

$$\boldsymbol{\mu}_{\mathsf{y}} := \mathbf{f}(\boldsymbol{\mu}_{\mathsf{x}}, \boldsymbol{\mu}_{\mathsf{n}}, \boldsymbol{\mu}_{\alpha}). \tag{15}$$

An important effect of linearising the mismatch function is that the corrupted speech turns out Gaussian. There are also other advantages that arise from fixing the expansion points, so that the relationship between speech, noise, and corrupted speech becomes linear per component. The means of the noise model can be estimated with a fixed-point iteration (Moreno, 1996) and the variance with a gradient-descent-based scheme (Liao, 2007). Alternatively, since the first-order approximation makes the noise, speech, and corrupted speech jointly Gaussian, an EM approach (Kim et al., 1998; Frey et al., 2001b; Kristjansson et al., 2001) can be used. However, the only way this can be done correctly is by assuming full covariance matrices for the noise and the noise-corrupted speech (Flego and Gales, 2011).

As discussed in section 2.1, previous work has assumed the phase factor 0 (Acero et al., 2000), or fixed to a different value, like 1 (Liao, 2007) or 2.5 (Li et al., 2007). A phase factor distribution has previously only been used for feature enhancement (Deng et al., 2004; Stouten et al., 2005; Yu et al., 2008). This work will apply VTS with a Gaussian phase factor for model compensation.

Compared to using a distribution, assuming $\boldsymbol{\alpha}$ constant has two effects. One is that the term $\mathbf{J}_\alpha \boldsymbol{\Sigma}_\alpha \mathbf{J}_\alpha^\mathsf{T}$ vanishes from the covariance expression in (14). Since the entries of the phase factor covariance are small, this decreases the variances only slightly. Since $\boldsymbol{\Sigma}_\alpha$ is constant across components and $\mathbf{J}_\alpha$ changes only slightly between adjacent components, discrimination is hardly affected. If $\boldsymbol{\alpha}$ is equal to its expected value, $\mathbf{0}$, then the mean of the compensated Gaussian does not change compared to when $\boldsymbol{\alpha}$ is assumed Gaussian. If $\boldsymbol{\alpha}$ is set to a higher value, the mean is overestimated. Also, Jacobians $\mathbf{J}_\mathsf{x}$ and $\mathbf{J}_\mathsf{n}$ move closer to $\frac{1}{2}\mathbf{I}$ (for more details, see van Dalen, 2011, appendix C.1).

In practice, $\boldsymbol{\alpha}$ is often assumed fixed but the noise model is estimated. This should subsume many of the effects that using a phase factor distribution would have had. This includes a wider compensated Gaussian and overestimation of the mode.

There are other ways of approximating the corrupted speech distribution with a Gaussian. One possibility is to approximate the mismatch function with a second-order vector Taylor series approximation (Stouten et al., 2005; Xu and Chin, 2009), and estimate the Gaussian to match the second moment of the resulting distribution. Another approximation is to use the unscented transformation (Julier and Uhlmann, 2004). This has been applied to feature enhancement (Shinohara and Akamine, 2009), and model compensation (Li et al., 2010; van Dalen and Gales, 2009). Yet another approach approximates the mismatch function with a piecewise linear approximation (Du and Huo, 2008; Seltzer et al., 2010), the parameters of which are learned from data. Whatever the differences between how these methods deal with the mismatch function, they all approximate the corrupted speech distribution with a diagonal-covariance Gaussian. When applying maximum-likelihood estimation to estimate the noise model, the differences between these compensation methods come down to slight variations in the parametrisation. For example, Li et al. (2010) finds that compensation with the VTS and the unscented transformation yields exactly the same performance when parameters for both are correctly optimised. Rather than looking into these variations, all of which produce Gaussian distributions, this paper will look into non-Gaussian distributions.

3.3. The Algonquin algorithm

Unlike the model compensation methods discussed so far, where distributions are explicitly estimated, the "Algonquin" algorithm (Frey et al., 2001a; Kristjansson and Frey, 2002) makes use of the observation. It is an extension to VTS compensation that uses a distinct expansion point for each observation. This is an important conceptual change as Algonquin specifically aims to yield the likelihood of an observation rather than the complete distribution. A given observation will be denoted with $\mathbf{y}_t$. Whereas VTS linearises the mismatch function at the expansion point given by the means of the prior distributions of the speech and the noise, the Algonquin algorithm updates the expansion point iteratively, intending to find the mode of the posterior of the speech and the noise. It can therefore be seen as an iterative approach to finding the Laplace approximation to the posterior.

The mismatch function does not explicitly take the phase factor into account, but adds uncertainty with variance $\boldsymbol{\Psi}$, so that

$$p(\mathbf{y}|\mathbf{x},\mathbf{n}) = \mathcal{N}(\mathbf{y};\, \mathbf{f}(\mathbf{x},\mathbf{n},\mathbf{0}),\, \boldsymbol{\Psi}) \simeq \mathcal{N}\Big(\mathbf{y};\, \mathbf{f}_{\mathsf{vts}}^{(k)}(\mathbf{x},\mathbf{n},\mathbf{0}),\, \boldsymbol{\Psi}\Big), \tag{16}$$

where $\mathbf{f}_{\mathsf{vts}}^{(k)}$ is the vector Taylor series approximation of the mismatch function at iteration k, with Taylor series expan-

sion point $(\mathbf{x}_0^{(k)}, \mathbf{n}_0^{(k)})$. Using the linearisation, $\mathbf{x}$, $\mathbf{n}$, and $\mathbf{y}$ are jointly Gaussian with

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{n} \\ \mathbf{y} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_{\mathsf{x}} \\ \boldsymbol{\mu}_{\mathsf{n}} \\ \boldsymbol{\mu}_{\mathsf{y}}^{(k)} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{\mathsf{x}} & \mathbf{0} & \boldsymbol{\Sigma}_{\mathsf{xy}}^{(k)} \\ \mathbf{0} & \boldsymbol{\Sigma}_{\mathsf{n}} & \boldsymbol{\Sigma}_{\mathsf{ny}}^{(k)} \\ \boldsymbol{\Sigma}_{\mathsf{yx}}^{(k)} & \boldsymbol{\Sigma}_{\mathsf{yn}}^{(k)} & \boldsymbol{\Sigma}_{\mathsf{y}}^{(k)} \end{bmatrix} \right). \tag{17}$$

When the expansion point changes, the speech and noise parameters do not change, but the approximation of the corrupted speech distribution $q_{\mathsf{y}_t}^{(k)} = \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_{\mathsf{y}}^{(k)}, \boldsymbol{\Sigma}_{\mathsf{y}}^{(k)})$ does. The speech and noise posteriors also change depending on the expansion point. It is well-known (see, e.g., Bishop, 2006, Appendix B) that from (17), their posterior distribution given the observation, $\mathbf{x}, \mathbf{n}|\mathbf{y}_t$, is Gaussian. The Algonquin algorithm sets the new expansion point $(\mathbf{x}_0^{(k)}, \mathbf{n}_0^{(k)})$ to the mean of this posterior. This yields a new approximate joint Gaussian (17). This process is then repeated for K iterations. Finally, the likelihood is computed by evaluating the observation-dependent Gaussian at the point given by the observation itself: $q_{\mathsf{y}_t}^{(K)}(\mathbf{y}_t)$.

For the likelihoods from the Algonquin algorithm to form a distribution, they need to be normalised. If $\mathbf{y}_t$ were kept fixed, then q_{y_t} would be Gaussian and thus normalised. However, for each observation $\mathbf{y}_t$, the Algonquin algorithm finds a different expansion point, and thus a different approximation q_{y_t}. This Gaussian is then evaluated at the same $\mathbf{y}_t$. Thus, in general, the likelihoods that the Algonquin algorithm finds are not normalised:

$$\int q_{\mathsf{y}_t}(\mathbf{y}_t)\, d\mathbf{y}_t \neq 1. \tag{18}$$

The original Algonquin algorithm does not require $q_{\mathsf{y}_t}(\mathbf{y}_t)$ to be a normalised distribution, because it finds a lower-bound approximation to the minimum mean square error estimate of the clean speech. This work, however, uses model compensation, which does require the distribution to be normalised. It may be possible to extend Algonquin model compensation with an empirical normalisation constant, but that is outside the scope of this paper. The Algonquin approximation will therefore not be used.

4. Asymptotically exact likelihood evaluation

This section will introduce a non-parametric approximation to the corrupted-speech likelihood, unlike the methods discussed in section 3, which find parametric distributions. The intuition underlying it is similar to that underlying the Algonquin algorithm (discussed in section 3.3). It is that no expression for the full density is needed: while recognising speech only the likelihood of vectors that are observed is required. Unlike the Algonquin algorithm, the integral that gives the likelihood of the corrupted speech (in (8c)) will be approximated directly. To approximate this integral, this work will apply importance sampling (a good introduction is Doucet and Johansen, 2008). Importance sampling requires a *proposal distribution*, the distribution that the algorithm draws from instead of the actual distribution. The proposal distributions needs to match the target density well, otherwise too many samples are required to arrive at a good estimate. The number of samples required grows exponentially with the number of dimensions.

Section 4.1 will write the corrupted speech likelihood as an integral over the speech $\mathbf{x}$ and the noise $\mathbf{n}$. To apply importance sampling, the proposal distribution will be either the prior, or an Algonquin-derived approximation of the posterior. Both are Gaussian joint distributions over the speech and the noise. Because the Gaussian cannot approximate the curved integrand well, the number of samples required makes it infeasible to use this approximation. Section 4.2 will then introduce a variable transformation of the integrand. This new expression is still exact, but more amenable to being approximated with importance sampling. After considering the one-dimensional case, this will be generalised to the higher-dimensional space, applying sequential importance re-sampling to approximate the integral dimension per dimension.

4.1. Importance sampling over the speech and the noise

This section presents the first approach to approximating the corrupted-speech likelihood, which is to approximate the integration over $\mathbf{x}$ and $\mathbf{n}$. The corrupted-speech likelihood was given in (8a), but here $\mathbf{y}_t$ will be substituted for $\mathbf{y}$. The likelihood expression can be written in two ways, implying two directions for Monte Carlo approximations.

$$p(\mathbf{y}_t) = \iint p(\mathbf{y}_t|\mathbf{x}, \mathbf{n})\, p(\mathbf{n}, \mathbf{x})\, d\mathbf{n}\, d\mathbf{x} = \iint \phi(\mathbf{x}, \mathbf{n}; \mathbf{y}_t) p(\mathbf{n}, \mathbf{x})\, d\mathbf{n}\, d\mathbf{x}. \tag{19a}$$

This expression looks like a standard Monte Carlo problem, where samples $(\mathbf{x}, \mathbf{n})$ must be drawn from their distributions, and the *test function* $\phi(\mathbf{x}, \mathbf{n}; \mathbf{y}_t)$ must be evaluated at these samples. The problem, however, is that for most values of $(\mathbf{x}, \mathbf{n})$, the test function, here equal to $p(\mathbf{y}_t|\mathbf{x}, \mathbf{n})$, has zero or very low values. For example, consider the case where the sample for the speech and the one for the noise have higher values than the observation that they are meant to explain. To make the Monte Carlo approximation feasible, it therefore must take into account the whole integrand, which will be written γ. The likelihood expression can therefore be written as

$$p(\mathbf{y}_t) = \iint p(\mathbf{y}_t|\mathbf{x}, \mathbf{n})\, p(\mathbf{n}, \mathbf{x})\, d\mathbf{n}\, d\mathbf{x} = \iint \gamma(\mathbf{x}, \mathbf{n}; \mathbf{y}_t)\, d\mathbf{n}\, d\mathbf{x} \triangleq Z. \tag{19b}$$

Here, Z is the *partition function*, or normalisation constant, of unnormalised density γ, so that $\iint \frac{1}{Z}\gamma(\mathbf{x}, \mathbf{n}; \mathbf{y}_t)\, d\mathbf{n}\, d\mathbf{x} = 1$. Most Monte Carlo methods, like Markov chain Monte Carlo, can deal with unnormalised densities, but cannot compute the normalisation constant. This section will therefore apply importance sampling, which is able to compute the normalisation constant.

Importance sampling requires that the integrand can be evaluated at any point $(\mathbf{x}, \mathbf{n})$. For this, $p(\mathbf{y}_t|\mathbf{x}, \mathbf{n})$ needs to be rewritten more explicitly. The observation $\mathbf{y}_t$ is not deterministic given the speech and the noise, because the phase factor $\boldsymbol{\alpha}$ is a random variable. However, if $\mathbf{x}$, $\mathbf{n}$, and $\mathbf{y}_t$ are given, $\boldsymbol{\alpha}$ is deterministic, so that $p(\mathbf{y}_t|\mathbf{x}, \mathbf{n})$ can be written in terms of the distribution of the phase factor. The phase factor that a specific setting of $\mathbf{x}$, $\mathbf{n}$, and $\mathbf{y}_t$ implies will be written $\boldsymbol{\alpha}(\mathbf{x}, \mathbf{n}, \mathbf{y}_t)$. It is a standard result (e.g. Bishop, 2006, 11.1.1) that transforming the space of a probability distribution requires multiplying by the determinant of the Jacobian:

$$p(\mathbf{y}_t|\mathbf{x}, \mathbf{n}) = \int \delta_{\mathbf{f}(\mathbf{x},\mathbf{n},\boldsymbol{\alpha})}(\mathbf{y}_t)\, p(\boldsymbol{\alpha})\, d\boldsymbol{\alpha} = \left| \frac{\partial \boldsymbol{\alpha}(\mathbf{x}, \mathbf{n}, \mathbf{y})}{\partial \mathbf{y}}\bigg|_{\mathbf{y}_t} \right| \cdot p(\boldsymbol{\alpha}(\mathbf{x}, \mathbf{n}, \mathbf{y}_t))\,, \tag{20}$$

where $p(\boldsymbol{\alpha}(\mathbf{x}, \mathbf{n}, \mathbf{y}))$ denotes the density of $p(\boldsymbol{\alpha})$ at the value of $\boldsymbol{\alpha}$ implied by $\mathbf{x}$, $\mathbf{n}$, and $\mathbf{y}$.

The value of the phase factor as a function of the other variables follows from (4). The relation is defined per coefficient (i.e. frequency bin) i of the variables:

$$\alpha_i = \frac{\exp(y_i) - \exp(x_i) - \exp(n_i)}{2\exp(\frac{1}{2}x_i + \frac{1}{2}n_i)}; \qquad \frac{\partial \alpha(x_i, n_i, y_i)}{\partial y_i} = \frac{\exp(y_i)}{2\exp(\frac{1}{2}x_i + \frac{1}{2}n_i)}. \tag{21}$$

The diagonal elements of the Jacobian in (20) are given by these partial derivatives with respect to y_i. The off-diagonal entries of the Jacobian are 0.

Then the distribution of the corrupted speech in (19b) can then be written as

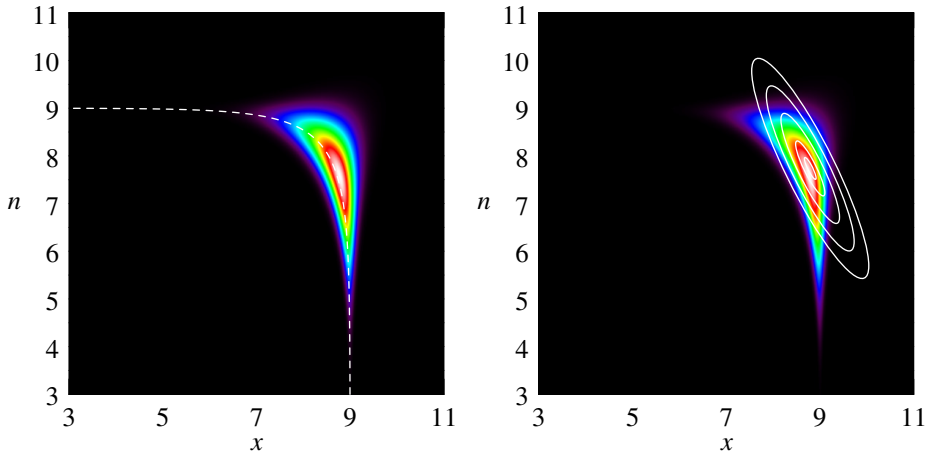
$$p(\mathbf{y}_t) = \iint \left[\left(\prod_i \frac{\exp(y_{t,i})}{2\exp(\frac{1}{2}x_i + \frac{1}{2}n_i)} \right) \cdot p(\boldsymbol{\alpha}(\mathbf{x}, \mathbf{n}, \mathbf{y}_t)) \right] p(\mathbf{n}, \mathbf{x})\, d\mathbf{n}\, d\mathbf{x}. \tag{22}$$

This expression is exact. Though the integrand is now straightforward to evaluate at any given point $(\mathbf{x}, \mathbf{n})$ if $p(\boldsymbol{\alpha})$ can be evaluated, the integral has no closed form. It can, however, be approximated using importance sampling. Figure 3a illustrates a one-dimensional version of the density $p(x, n, y_t) = \gamma(x, n; y_t)$. The density lies around the curve that relates x and n for given α and y_t, shown as a dashed line. This curve is given by $\exp(x) + \exp(n) = \exp(y_t)$. The constraint that the sum of the exponents of x and n is fixed causes the curve to be bent.

Importance sampling uses a proposal distribution ρ, from which L samples $(\mathbf{x}^{(l)}, \mathbf{n}^{(l)})$ are drawn. These are weighted to reflect the difference between proposal and target densities. Thus the integral in (22) can be approximated as

$$\iint \gamma(\mathbf{x}, \mathbf{n}; \mathbf{y}_t)\, d\mathbf{n}\, d\mathbf{x} = \iint \frac{\gamma(\mathbf{x}, \mathbf{n}; \mathbf{y}_t)}{\rho(\mathbf{x}, \mathbf{n})} \rho(\mathbf{x}, \mathbf{n})\, d\mathbf{n}\, d\mathbf{x} \simeq \sum_{l=1}^{L} \frac{\gamma(\mathbf{x}^{(l)}, \mathbf{n}^{(l)}; \mathbf{y}_t)}{\rho(\mathbf{x}^{(l)}, \mathbf{n}^{(l)})}, \qquad (\mathbf{x}^{(l)}, \mathbf{n}^{(l)}) \sim \rho. \tag{23}$$

The ratio of the target and proposal densities, γ/ρ, gives the weight of each sample. In this section, a Gaussian proposal distribution is considered: the Algonquin approximation to the posterior (see section 3.3). A priori, the speech and the noise are independent, but the Algonquin-estimated Gaussian is not. Figure 3b shows it superimposed on the actual posterior.



(a) The distribution $p(x, n, y_t = 9)$.

(b) Gaussian approximation with Algonquin.

Figure 3: The distribution of the clean speech and noise for $x \sim \mathcal{N}(10, 1)$; $n \sim \mathcal{N}(9, 2)$; $\alpha \sim \mathcal{N}(0, 0.04)$; $y_t = 9$.

The main problem areas for any Gaussian proposal distribution are the regions of space where proposal and target do not match well. Where the proposal distribution has a higher value than the target distribution, more samples will be drawn that are assigned lower weights: a waste of computational time. Conversely, where the proposal distribution is much lower than the target distribution, samples will seldom be drawn, and when they do, they are assigned high weights. In this case, the number of samples that needs to be drawn to get sufficient coverage becomes very high.

These two problems are exacerbated by high dimensionality: for every dimension, either of these cases can occur. Therefore, the number of samples required increases exponentially with the dimensionality. The Algonquin approximation illustrated in figure 3b suffers from this problem, so that it is not feasible to apply it to a 24-dimensional log-spectral problem. Instead, the next section will therefore transform the space so that the target distribution per dimension can be approximated better.

4.2. Transformed-space sampling

The problem with the scheme in the previous section is the difficulty in obtaining a proposal function that is a good approximation to the distribution of x and n when y_t is given. This section will overcome this by transforming the space of the integral. Conceptually, this is similar to Myrvoll and Nakamura (2004), where the integrand was approximated with line segments. However, in that work the approximation was constrained to one dimension. Generalising this to multiple dimensions is possible in theory (see van Dalen, 2011, appendix E.2), but results in a summation over a number of hyperplanes exponential in the number of dimensions. Additionally, the mismatch function in this work contains an extra variable (the phase factor). Thus a modified transformation is required. Initially, a one-dimensional space will be considered. Then it will be discussed how to generalise this to the multi-dimensional case and how to deal with the additional complications that arise.

Single-dimensional

In one dimension, the mismatch function is ((4) without indices and with y_t substituted for y)

$$\exp(y_t) = \exp(x) + \exp(n) + 2\alpha \exp\left(\tfrac{1}{2}x + \tfrac{1}{2}n\right). \tag{24}$$

The objective of this section is to find an approximation to $p(y_t)$ for a given observation y_t. Since the four variables in (24) are linked deterministically and one is known (y_t), the integration will be over two variables. This was also the case in section 4.1. There, the choice of integrating over x and n did not work because of the complexity of the density in (x, n)-space. This section introduces a variable u that represents a pair (x, n) given y_t and α. Changing u traverses

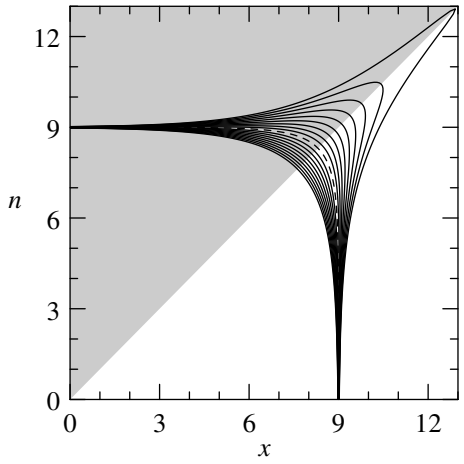


Figure 4: The region of the (x, n) that the integral is explicitly derived for: $x \leq n$.

the curve in (x, n)-space, which removes the bend (the dashed line in figure 3). The integral will then be over α and the new variable u. The substitute variable is defined as

$$u = n - x. \tag{25}$$

As x and n are in the log-spectral domain, the value of u is related to the signal-to-noise ratio. If u is a large negative number, x is close to y_t and n is large and negative. This represents a very low signal-to-noise ratio. The converse is true if u is a large positive number: then n is close to y_t and x is large and negative.

This substitution will be used to define an integral that yields $p(y_t)$. First, $p(y_t|\cdot)$ will be re-expressed using $p(n|\cdot)$ or $p(x|\cdot)$. Since neither of these variables is known deterministically for all values of the other variables, the integral will be partitioned in two parts. n has one possible value given a setting for (x, α, y_t) when x is constrained to be smaller than n, which is the shaded region in figure 4. In the complementary region, x has one possible value given fixed (n, α, y_t). The likelihood can be written as a sum of both these regions:

$$p(y_t) = p(y_t, x \leq n) + p(y_t, n < x)\,. \tag{26}$$

Because of the symmetry between these two regions, only the derivation for the region $x \leq n$ will be given explicitly. The derivation for $n < x$ is analogous.

The value of the additive noise n that follows from setting the variables x, α, y_t will be denoted with $n(x, \alpha, y_t)$. This is deterministic in the region where $x \leq n$. The expression, straightforward to derive (see van Dalen, 2011, appendix F.1), is:

$$n(x, \alpha, y_t) = 2\log\left(-\alpha \exp\left(\tfrac{1}{2}x\right) + \sqrt{\exp(y_t) + \exp(x)\,(\alpha^2 - 1)}\right). \tag{27}$$

The variable of the probability distribution will be changed from y_t to n. This requires multiplication by a Jacobian (see, for example, Bishop, 2006, 11.1.1). This Jacobian, the partial derivative of n with respect to y and keeping x and α fixed, will be written $\frac{\partial n(x,\alpha,y)}{\partial y}$, and be evaluated at y_t.

$$p(y_t, x \leq n|x, \alpha) = \left|\left.\frac{\partial n(x, \alpha, y)}{\partial y}\right|_{y_t}\right| \cdot 1(x \leq n) \cdot p(n(x, \alpha, y_t))\,. \tag{28}$$

Here, $1(\cdot)$ denotes the indicator function, which evaluates to 1 if its argument is true, and 0 otherwise. $p(n(x, \alpha, y_t))$ is the probability distribution of n evaluated at $n(x, \alpha, y_t)$, the value of n corresponding to the setting of (x, α, y_t).

The evaluation of the half of the likelihood for $x \leq n$ can then be rewritten with (28) and by then replacing the variable of the integration by u. The predicate $x \leq n$ is equivalent to $0 \leq u$, which can be expressed using bounds on

the integral.

$$\begin{aligned} p(y_t, x \le n) &= \int p(\alpha) \int p(x)\, p(y_t, x \le n | x, \alpha)\, dx\, d\alpha \\ &= \int p(\alpha) \int p(x) \cdot \left| \frac{\partial n(x,\alpha,y)}{\partial y}\bigg|_{y_t} \right| \cdot 1(x \le n) \cdot p(n(x,\alpha,y_t))\, dx\, d\alpha \\ &= \int p(\alpha) \int_0^\infty \left| \frac{\partial x(u,\alpha,y_t)}{\partial u} \right| \cdot \left| \frac{\partial n(x,\alpha,y)}{\partial y}\bigg|_{y_t, x(u,\alpha,y_t)} \right| \cdot p(x(u,\alpha,y_t)) \cdot p(n(u,\alpha,y_t))\ du\, d\alpha. \end{aligned} \tag{29}$$

Here, $p(x(u,\alpha,y_t))$ is the probability distribution of x evaluated at $x(u,\alpha,y_t)$, the value of x corresponding to the setting of (u,α,y_t), and similar for $p(n(u,\alpha,y_t))$. These values are given by

$$x(u,\alpha,y_t) = y_t - \log\Big(1 + \exp(u) + 2\alpha \exp\Big(\tfrac{1}{2}u\Big)\Big); \qquad n(u,\alpha,y_t) = y_t - \log\Big(1 + \exp(-u) + 2\alpha \exp\Big(-\tfrac{1}{2}u\Big)\Big). \tag{30}$$

Van Dalen (2011, appendix F.1) gives the derivations for these, and for the Jacobians in (29), for the region $x \le n$. The product of the Jacobians is -1. By substituting 1 for the absolute value of the product of the Jacobians into (29), one half of the likelihood in (26) becomes

$$p(y_t, x \le n) = \int p(\alpha) \int_0^\infty p(x(u,\alpha,y_t))\, p(n(u,\alpha,y_t))\, du\, d\alpha. \tag{31a}$$

The integral of u is over the area where $0 \le u$, i.e. where $x \le n$. The equivalent integration over the region where $u < 0$ could be derived by exchanging x and n, and replacing u with $-u$. Applying this to (31a) yields the other half of the likelihood in (26). Because of the symmetry of n and x and because the Jacobians cancel out, the result is identical to (31a) save for the range of u:

$$p(y_t, n < x) = \int p(\alpha) \int_{-\infty}^0 p(x(u,\alpha,y_t))\, p(n(u,\alpha,y_t))\, du\, d\alpha. \tag{31b}$$

The sum of (31a) and (31b) yields the total likelihood of y_t. The integrand will be called γ.

$$p(y_t) = p(y_t, x \le n) + p(y_t, n < x) = \int p(\alpha) \int_{-\infty}^\infty p(x(u,\alpha,y_t))\, p(n(u,\alpha,y_t))\, du\, d\alpha = \iint \gamma(u,\alpha;y_t)\, du\ d\alpha, \tag{32a}$$

where

$$\gamma(u,\alpha;y_t) \triangleq p(\alpha)\, \gamma(u;\alpha,y_t), \qquad \gamma(u;\alpha,y_t) \triangleq p(x(u,\alpha,y_t))\, p(n(u,\alpha,y_t)). \tag{32b}$$

Thus the integral has been expressed in terms of u and α, instead of x and n as in (22). This derivation is exact and holds for any form of priors for the speech and noise $p(x)$ and $p(n)$. The integrand can be evaluated at any given point (u,α), assuming that $p(\alpha)$ can be evaluated, but the integral has no closed form. The terms in $\gamma(u;\alpha,y_t)$ are two Gaussians with differently-transformed variables. The outer integral is straightforward to approximate with standard Monte Carlo by sampling $\alpha^{(l)}$ from $p(\alpha)$. The problem with approximating the inner integral is that it is not possible to draw samples from $\gamma(u;\alpha,y_t)$. Therefore, importance sampling is necessary. This requires a proposal distribution $\rho(u|\alpha)$ that it is possible to draw samples from, and is close to γ. Intuitively, the double integral can then be replaced by a summation over L samples $\alpha^{(l)}$ from $p(\alpha)$ and corresponding samples for $u^{(l)}$ drawn from $\rho(u|\alpha^{(l)})$:

$$\begin{aligned} \iint \gamma(u;\alpha,y_t)\, du\, p(\alpha)\, d\alpha &= \iint \frac{\gamma(u;\alpha,y_t)}{\rho(u|\alpha)} \rho(u|\alpha)\, du\, p(\alpha)\, d\alpha \\ &\simeq \frac{1}{L} \sum_{l=1}^L \int \frac{\gamma(u;\alpha^{(l)},y_t)}{\rho(u|\alpha^{(l)})} \rho(u|\alpha^{(l)})\, du \simeq \frac{1}{L} \sum_{l=1}^L \frac{\gamma(u^{(l)};\alpha^{(l)},y_t)}{\rho(u^{(l)}|\alpha^{(l)})}, \quad \alpha^{(l)} \sim p(\alpha), \quad u^{(l)} \sim \rho(u|\alpha^{(l)}). \end{aligned} \tag{33a}$$

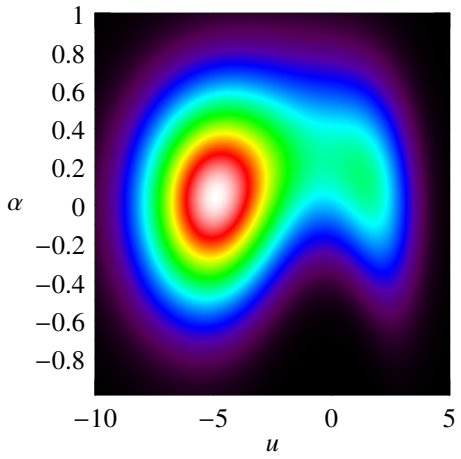
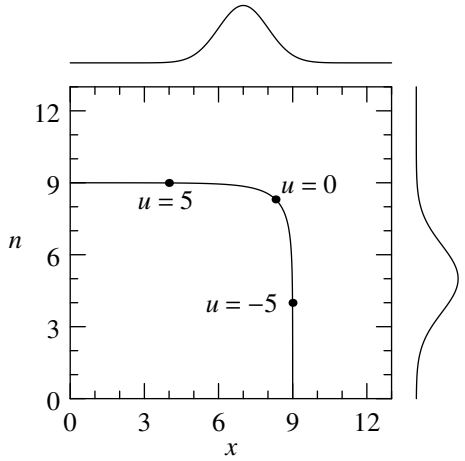


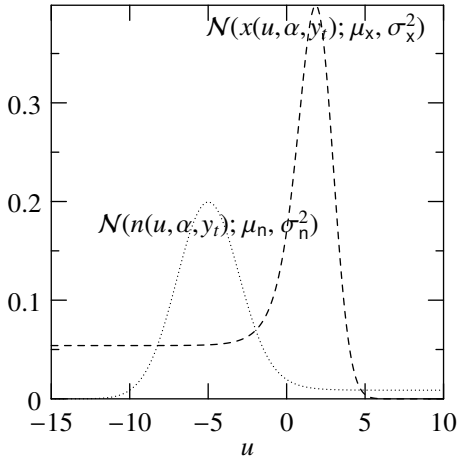
Figure 5: The density of $\gamma(u, \alpha; y_t) = p(\alpha)\,\gamma(u; \alpha, y_t)$ for $y_t = 9, x \sim \mathcal{N}(7, 1), n \sim \mathcal{N}(4, 4), \sigma_\alpha^2 = 0.13$.

The shape of the integrand. To apply importance sampling, the proposal distribution $\rho(u|\alpha^{(l)})$ needs to be tailored to the integrand. As discussed in section 4.1, it is important that the proposal distribution matches the integrand closely, or too many samples will be required for a good approximation. This section will find proposal distributions with well-matching shapes. The scaling of the proposal distribution graphs in this section will be arbitrary.

In this work, the speech and noise Gaussians have full covariance matrices (as in (1)) as the mismatch function is defined in the log-spectral domain. Figure 5 gives an example of the integrand $\gamma(u, \alpha; y_t)$. Samples can be directly drawn from the distribution for α. Thus it is only necessary to find a proposal function for $\gamma(u; \alpha, y_t)$ to draw samples for u. The following examples will assume the mode of $p(\alpha)$, $\alpha = 0$, and consider representative shapes for $\gamma(u; \alpha = 0, y_t)$.



(a) x, n for fixed $y_t = 9, \alpha = 0$, and varying u. At the top $\mathcal{N}(x;\, \mu_{\mathsf{x}},\, \sigma_{\mathsf{x}}^2)$ in γ; right $\mathcal{N}(n;\, \mu_{\mathsf{n}},\, \sigma_{\mathsf{n}}^2)$.



(b) The factors of $\gamma(u; \alpha = 0, y_t)$ separately. For $y_t = 9$, $x \sim \mathcal{N}(7, 1)$ (– –), $n \sim \mathcal{N}(4, 4)$ $(\cdots)$.

Figure 6: The shape of the integrand in one dimension

Figure 6a depicts the relationship between x and n. Different values of u represent different positions on the curve. The two factors of $\gamma(u; \alpha, y_t)$ are the Gaussians depicted on top and on the side of the graph. $\gamma(u; \alpha, y_t)$ consists of a factor $\mathcal{N}(x(u, \alpha, y_t);\, \mu_{\mathsf{x}},\, \sigma_{\mathsf{x}}^2)$ related to the clean speech and a factor $\mathcal{N}(n(u, \alpha, y_t);\, \mu_{\mathsf{n}},\, \sigma_{\mathsf{n}}^2)$ related to the noise. Both terms are Gaussians, but the variables of the Gaussians (x and n) are non-linear functions of u. This graph provides

an intuitive connection to noise masking schemes, which assume that either the speech or the noise dominates (Klatt, 1976; Holmes and Sedgwick, 1986). This would yield a curve with a sharp angle, so that always either the speech or the noise equals the observation.

Figure 6b illustrates the shape of the two factors. When the Gaussians are plotted with respect to u, the soft cut-off leads to a Gaussian-like distribution that is infinitely extended on one side. As u tends to $-\infty$, x tends to y_t. In this example, as $u \to -\infty$, $\mathcal{N}(x(u, \alpha, y_t);\, \mu_{\mathsf{x}},\, \sigma_{\mathsf{x}}^2)$ therefore tends to

$$\mathcal{N}(x(u, \alpha, y_t);\, \mu_{\mathsf{x}},\, \sigma_{\mathsf{x}}^2) \to \mathcal{N}(y_t;\, \mu_{\mathsf{x}},\, \sigma_{\mathsf{x}}^2) = \mathcal{N}(9;\, 7,\, 1) = e^{-2} / \sqrt{2\pi} \simeq 0.054. \tag{34}$$

Analogously, $\mathcal{N}(n(u, \alpha, y_t);\, \mu_{\mathsf{n}},\, \sigma_{\mathsf{n}}^2)$ is Gaussian-like but cut off at its right tail, where it converges to a non-zero constant.

Importance sampling from the integrand. To apply importance sampling, a proposal distribution must be found for each $\gamma(u; \alpha, y_t)$. Samples will be drawn from this proposal distribution ρ, and then weighted by γ/ρ. The proposal distribution should be similar to the actual integrand. Most importantly, it should have mass everywhere the integrand has mass. If the integrand is not completely covered, then few samples can be assigned arbitrarily high weights, which creates large and unbounded variances in the quantity to be estimated. If, conversely, the proposal distribution has mass where the integrand has hardly any, some sample receive weights close to 0, which has only a bounded effect on the variance of the final estimate. Proposal distributions are of a heuristic nature; if it were possible to mathematically guarantee coverage, it would most likely be possible to draw samples from the integrand directly.

A proposal distribution that it is straightforward to draw a sample from is a Gaussian mixture model. Figure 7 shows examples for the three types of shape of $\gamma(u; \alpha, y_t)$. The shape of the factors γ is different depending on whether the speech and noise are less or greater than the observation. Therefore, the approximation will be different for each of these cases. The magnitudes of the proposal distributions are scaled to make the areas under the integrand and the proposal density equal. The proposal distributions must be defined over u, and will require the value of u corresponding to a specific setting of x, α, and y_t (and n, α, and y_t). This will be denoted $u_x(x, \alpha, y_t)$ (and $u_n(n, \alpha, y_t)$). The expressions are derived in van Dalen (2011, appendix F.1). The following yields $u > 0$, which lies in the shaded region in figure 4:

$$u_x(x, \alpha, y_t) = 2 \log\left(-\alpha + \sqrt{\exp(y_t - x) + \alpha^2 - 1}\right). \tag{35a}$$

The mirror image of this expression is $u_n(n, \alpha, y_t)$, which fixes n rather than x, and yields $u < 0$.

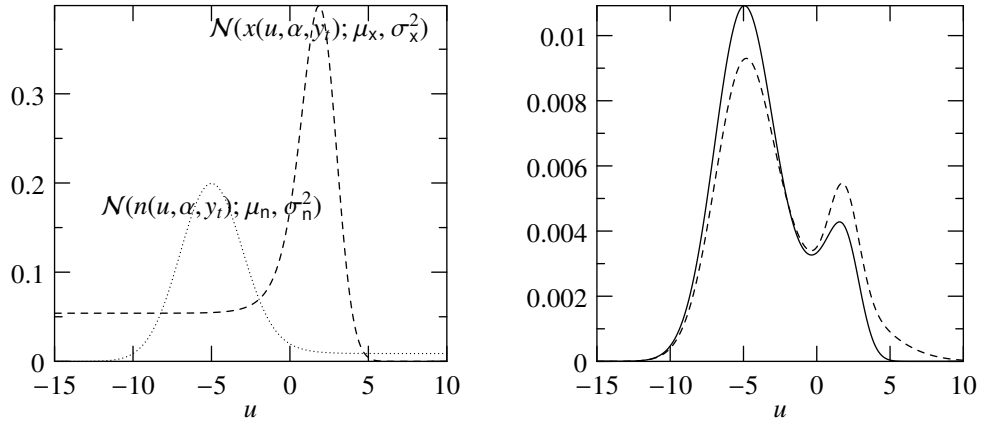
$$u_n(n, \alpha, y_t) = -2 \log\left(-\alpha + \sqrt{\exp(y_t - n) + \alpha^2 - 1}\right). \tag{35b}$$

The proposal distribution is chosen differently depending on the mean of the terms of γ as follows:

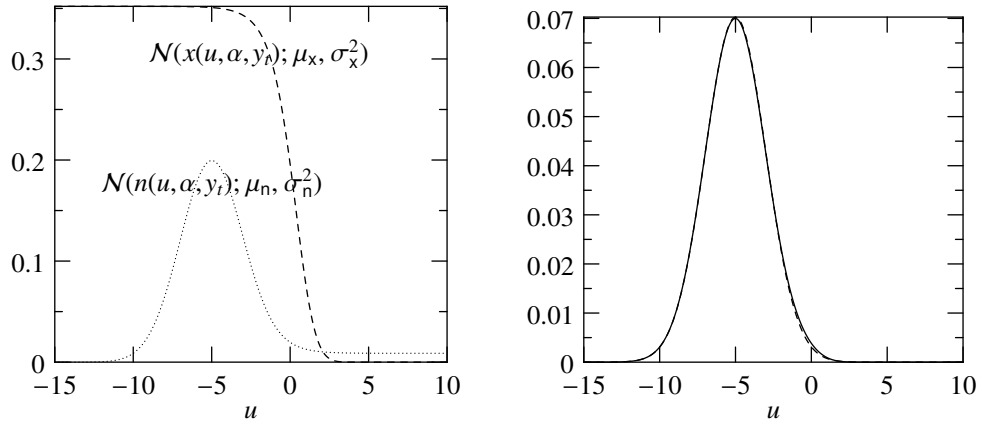
1. $\mu_{\mathsf{x}} < y_t$ and $\mu_{\mathsf{n}} < y_t$. This produces a shape of γ as in figure 7a. Figure 7a uses the example from figure 6b. The integrand, the product of two cut-off Gaussians, is bimodal. When u tends to $\pm\infty$, one term of γ tends to a non-zero constant, but the other one tends to 0. γ therefore tends to 0 as well.[1] The shape of γ is close to the mixture of two Gaussians.
   A mixture of two Gaussians therefore seems a reasonable proposal distribution, with means at the approximate modes, and covariances set to σ_{x}^2 and σ_{n}^2, respectively. They approximate the terms $\mathcal{N}(n(u, \alpha, y_t);\, \mu_{\mathsf{n}},\, \sigma_{\mathsf{n}}^2)$ with $\mathcal{N}(u;\, u_n(\mu_{\mathsf{n}}, \alpha, y_t),\, \sigma_{\mathsf{n}}^2)$ and $\mathcal{N}(x(u, \alpha, y_t);\, \mu_{\mathsf{x}},\, \sigma_{\mathsf{x}}^2)$ with $\mathcal{N}(u;\, u_x(\mu_{\mathsf{x}}, \alpha, y_t),\, \sigma_{\mathsf{x}}^2)$. As was seen in figure 6b, each Gaussian is essentially scaled by the extended tail of the other one. The weights of the Gaussians of the proposal distribution can be set to the corresponding value of the tail of the other one. The mode of the term related to the speech in figure 6b is approximately at $u_x(\mu_{\mathsf{x}}, \alpha, y_t)$. The component of the proposal distribution related to this term is therefore multiplied by the value of the term related to the noise at this point, and vice versa:

$$\omega_{\mathsf{x}} \propto \mathcal{N}(n(u_x(\mu_{\mathsf{x}}, \alpha, y_t), \alpha, y_t);\, \mu_{\mathsf{n}},\, \sigma_{\mathsf{n}}^2); \qquad \omega_{\mathsf{n}} \propto \mathcal{N}(x(u_n(\mu_{\mathsf{n}}, \alpha, y_t), \alpha, y_t);\, \mu_{\mathsf{x}},\, \sigma_{\mathsf{x}}^2). \tag{36a}$$

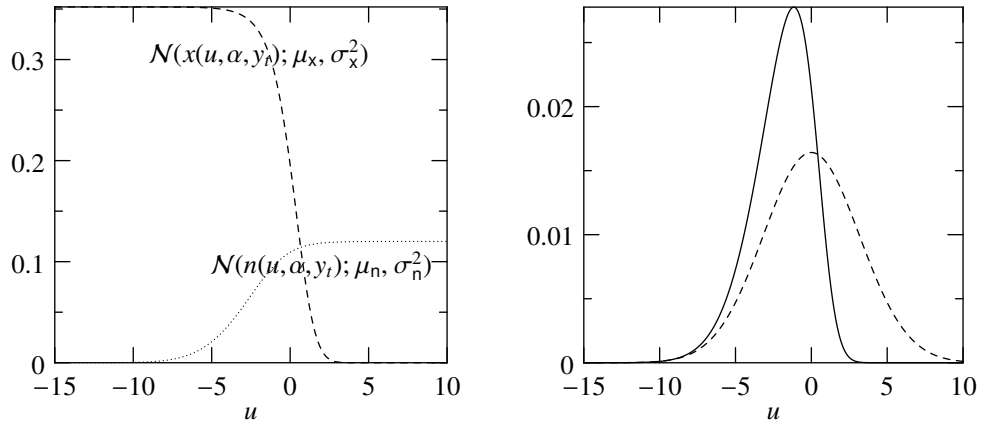
[1] This must be true also because $\int \gamma(u; \alpha, y_t)\, du$ is equal to $p(y_t|\alpha)$ evaluated at a point, which cannot be infinite if either σ_{x}^2 or σ_{n}^2 is non-zero.



(a) The factors of $\gamma(u;\alpha=0,y_t=9)$: $\mathcal{N}(x(u,\alpha,y_t);\ 7,\ 1)$ (– –); $\mathcal{N}(n(u,\alpha,y_t);\ 4,\ 4)$ ($\cdots$). Their product $\gamma(u;\alpha,y_t)$ (—) and the proposal distribution $\rho(u|\alpha)$ (– –).



(b) The factors of $\gamma(u;\alpha=0,y_t=9)$: $\mathcal{N}(x(u,\alpha,y_t);\ 9.5,\ 1)$ (– –); $\mathcal{N}(n(u,\alpha,y_t);\ 4,\ 4)$ ($\cdots$). Their product $\gamma(u;\alpha,y_t)$ (—) and the proposal distribution $\rho(u|\alpha)$ (– –).



(c) The factors of $\gamma(u;\alpha=0,y_t=9)$: $\mathcal{N}(x(u,\alpha,y_t);\ 9.5,\ 1)$ (– –); $\mathcal{N}(n(u,\alpha,y_t);\ 10,\ 10)$ ($\cdots$). Their product $\gamma(u;\alpha,y_t)$ (—) and the proposal distribution $\rho(u|\alpha)$ (– –).

Figure 7: $\gamma(u;\alpha,y_t)$ for different cases. Left the two factors. Right the integrand with the proposal distribution.

However, this can cause the area between the two peaks to be under-covered. A third Gaussian is therefore added in the middle of the peaks. Its variance σ_3^2 is set to the variance of the means of the other two Gaussians, $\frac{1}{4}(u_x(\mu_\mathsf{x}, \alpha, y_t) - u_n(\mu_\mathsf{n}, \alpha, y_t))^2$, so that it covers the area between the means. Its weight so that the height of the Gaussian equals the value of the integrand at $u = 0$:

$$\omega_3 \propto \gamma(0; \alpha, y_t)\sqrt{2\pi\sigma_3^2}, \tag{36b}$$

where the square root is a term that compensates for the height of the Gaussian. ω_3 thus replaces the normalisation constant of the Gaussian by $\gamma(0; \alpha, y_t)$.
These three weights are normalised so that they sum to 1. The distribution becomes

$$\rho(u) = \omega_\mathsf{n}\mathcal{N}(u;\, u_n(\mu_\mathsf{n}, \alpha, y_t),\, \sigma_\mathsf{n}^2) + \omega_\mathsf{x}\mathcal{N}(u;\, u_x(\mu_\mathsf{x}, \alpha, y_t),\, \sigma_\mathsf{x}^2) + \omega_3\mathcal{N}(u;\, 0,\, \sigma_3^2). \tag{36c}$$

2. $\mu_\mathsf{x} > y_t$ and $\mu_\mathsf{n} < y_t$ (or its mirror image, $\mu_\mathsf{x} < y_t$ and $\mu_\mathsf{n} > y_t$). Figure 7b shows that $\mathcal{N}(x(u, \alpha, y_t);\, \mu_\mathsf{x},\, \sigma_\mathsf{x}^2)$ is cut off before its peak, and converges to its maximum in the limit as $u \to -\infty$. This results in a Gaussian distribution except for one tail. The proposal distribution is therefore set to this Gaussian:

$$\rho(u) = \mathcal{N}(u;\, u_n(\mu_\mathsf{n}, \alpha, y_t),\, \sigma_\mathsf{n}^2). \tag{37}$$

   The right-hand graph of figure 7b shows the near-perfect match of this proposal distribution.
3. $\mu_\mathsf{n} > y_t$ and $\mu_\mathsf{x} > y_t$. Both terms of γ are cut off before their peaks, resulting in a shape as in figure 7c. The product is a distribution around $u = 0$ with Gaussian-like tails, one derived from $\mathcal{N}(n(u, \alpha, y_t);\, \mu_\mathsf{n},\, \sigma_\mathsf{n}^2)$ and another one derived from $\mathcal{N}(x(u, \alpha, y_t);\, \mu_\mathsf{x},\, \sigma_\mathsf{x}^2)$. The proposal distribution is therefore set to a Gaussian with mean 0. Its variance is set to the largest of the variances of the speech and the noise:

$$\rho(u) = \mathcal{N}(u;\, 0,\, \max(\sigma_\mathsf{n}^2, \sigma_\mathsf{x}^2)). \tag{38}$$

   As the right-hand graph in figure 7c shows, this provides good coverage but over-estimation on part of the space. This means that some samples will receive a very low weight, but that is not a problem.

Thus, by transforming the space of the integration from (x, n) to (u, α), much better proposal distributions for importance sampling can be found than in (x, n)-space, like in section 4.1. The sample weights will therefore vary less, so that good approximations to the integral will be found with a much smaller number of samples.

Special cases of either case (1) or (2) is when there is very little or no noise. Then, $\mu_\mathsf{n} \ll y_t$ or $\mu_\mathsf{n} \to -\infty$. In either case, both the integrand and the proposal distribution become dominated by the Gaussian resulting from the noise. The only difference between the integrand and the proposal is the height of this Gaussian, which is determined by the value of the clean speech Gaussian at y_t, as in (34). This value becomes the weight of any sample drawn from the proposal distribution. This means that in the trivial case where there is no noise, with the choice of proposal distribution proposed above only one sample is necessary to evaluate the corrupted-speech likelihood exactly.

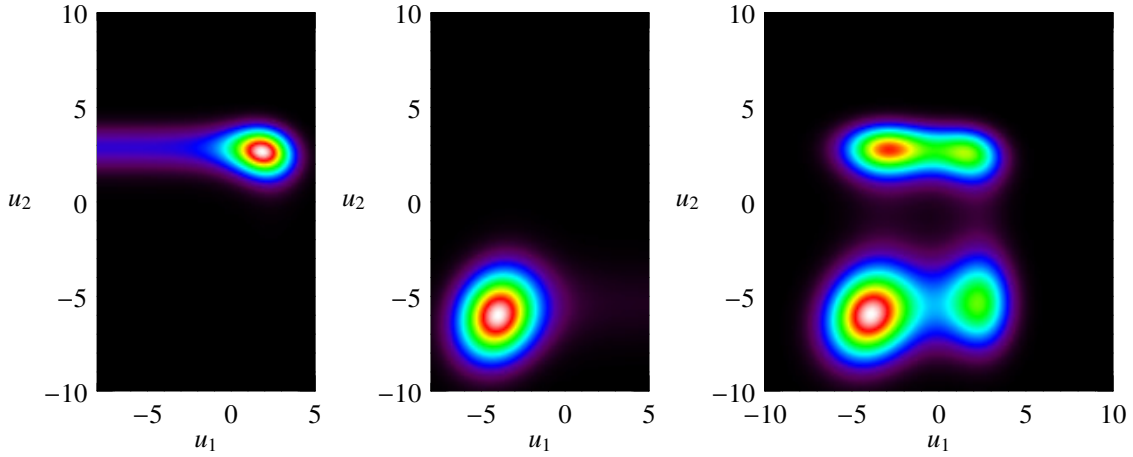
The next section will extend the techniques applied in this chapter to the multi-dimensional case.

Multi-dimensional

So far only a single dimension has been considered. The relations between the single-dimensional variables in the previous section hold per dimension for the multi-variate case. The substitute variable $\mathbf{u}$ that is introduced to represent a point $(\mathbf{x}, \mathbf{n})$ given observation $\mathbf{y}_t$ and phase factor vector $\boldsymbol{\alpha}$ is therefore defined as

$$\mathbf{u} = \mathbf{n} - \mathbf{x}. \tag{39}$$

There was a complication in transforming the one-dimensional integral in section 4.2 from (x, n) to (u, α): for some x, multiple values for n were possible, and vice versa. Because transforming the integral needed a deterministic link, the integral was split into two parts, for two regions of (x, n)-space. In the multi-dimensional case it is necessary to do this for each of the dimensions. Van Dalen (2011, appendix F.2) gives the full derivation. The integration is therefore first split up into conditional distributions per dimension i, and then into regions. The integrals for the two



(a) $\mathcal{N}(\mathbf{x}(\mathbf{u}, \boldsymbol{\alpha}, \mathbf{y}_t); \boldsymbol{\mu}_\mathsf{x}, \boldsymbol{\Sigma}_\mathsf{x})$. (b) $\mathcal{N}(\mathbf{n}(\mathbf{u}, \boldsymbol{\alpha}, \mathbf{y}_t); \boldsymbol{\mu}_\mathsf{n}, \boldsymbol{\Sigma}_\mathsf{n})$. (c) The product: $\gamma(\mathbf{u}; \boldsymbol{\alpha}, \mathbf{y}_t)$.

Figure 8: The integrand $\gamma(\mathbf{u}; \boldsymbol{\alpha}, \mathbf{y}_t)$ for $\boldsymbol{\alpha} = \mathbf{0}$: the two factors, and their product.

regions over (x_i, n_i) are rewritten as an integral over (u_i, α_i). Collapsing the dimensions then yields an unsurprising generalisation of (32a):

$$p(\mathbf{y}_t) = \int p(\boldsymbol{\alpha}) \int p(\mathbf{x}(\mathbf{u}, \boldsymbol{\alpha}, \mathbf{y}_t))\, p(\mathbf{n}(\mathbf{u}, \boldsymbol{\alpha}, \mathbf{y}_t))\, d\mathbf{u}\, d\boldsymbol{\alpha} = \iint \gamma(\mathbf{u}, \boldsymbol{\alpha}; \mathbf{y}_t)\, d\mathbf{u}\, d\boldsymbol{\alpha}, \tag{40a}$$

where

$$\gamma(\mathbf{u}, \boldsymbol{\alpha}; \mathbf{y}_t) \triangleq p(\boldsymbol{\alpha})\, \gamma(\mathbf{u}; \boldsymbol{\alpha}, \mathbf{y}_t)\,; \qquad \gamma(\mathbf{u}; \boldsymbol{\alpha}, \mathbf{y}_t) \triangleq p(\mathbf{x}(\mathbf{u}, \boldsymbol{\alpha}, \mathbf{y}_t))\, p(\mathbf{n}(\mathbf{u}, \boldsymbol{\alpha}, \mathbf{y}_t))\,. \tag{40b}$$

This derivation is valid whatever the form of the speech and noise priors, $p(\mathbf{x})$ and $p(\mathbf{n})$. To approximate this integral, conventional importance sampling could again be used, if the dimensions were not correlated due to the full covariance matrices.

Figure 8 illustrates how the shape of the integrand $\gamma(\mathbf{u}; \boldsymbol{\alpha} = \mathbf{0}, \mathbf{y}_t)$ generalises to two dimensions of $\mathbf{u}$. The principles are the same as the one-dimensional case in figure 6. Figure 8a shows the factor of γ deriving from the speech prior, $\mathcal{N}(\mathbf{x}(\mathbf{u}, \boldsymbol{\alpha}, \mathbf{y}_t); \boldsymbol{\mu}_\mathsf{x}, \boldsymbol{\Sigma}_\mathsf{x})$, and figure 8b the same for the noise prior. They are again Gaussian-like with a soft cut-off, this time in two directions. By choosing the parameter setting

$$\mathbf{x} \sim \mathcal{N}\left(\begin{bmatrix} 7 \\ 6.3 \end{bmatrix}, \begin{bmatrix} 1 & -0.1 \\ -0.1 & 0.5 \end{bmatrix}\right); \qquad \mathbf{n} \sim \mathcal{N}\left(\begin{bmatrix} 5 \\ 3 \end{bmatrix}, \begin{bmatrix} 2 & 0.3 \\ 0.3 & 2 \end{bmatrix}\right); \qquad \mathbf{y}_t = \begin{bmatrix} 9 \\ 9 \end{bmatrix}, \tag{41}$$

the product of the two factors, in figure 8c, turns out to have four maxima. In general, for d dimensions, the integrand can have 2^d modes. A mixture of Gaussians could be again used for the proposal distribution, but would need as many components as the integrand has maxima. Therefore, instead of applying normal importance sampling, the integrand will be factorised in dimensions for sequential importance sampling.

Sequential importance sampling (Kitagawa, 1996) is an approach for multi-dimensional sampling. First, a set of samples are drawn from a distribution over the first dimension, and each sample assigned a weight. Then, for each dimension every partial sample is extended with a value drawn given the value for previous dimensions of that sample. The advantage of this formulation is that between dimensions it allows for re-sampling: duplicating some samples from the set and removing other ones. This concentrates the samples on the higher-probability areas.

To be able to apply sequential importance sampling, the integrand needs to be factorised into dimensions. These will be referred to as factors. If the feature space is d-dimensional, the integration is over $2d$ dimensions: $\alpha_1, \ldots, \alpha_d$, $u_1, \ldots, u_d$. The factors do not need to represent conditional probability distributions. It is true that the most informative

weights after each dimension i would arise if factors $1 \dots i$ were combined to form the (potentially unnormalised) marginal of partial sample $u_{1:i}$. Re-sampling would then be most effective. By definition, the factors would be (unnormalised) conditionals. However, this is not a requirement; see van Dalen (2011, 7.3.2) for a comparison between two different factorisations.

In this work, $\gamma(\mathbf{u}; \boldsymbol{\alpha}, \mathbf{y}_t)$ is the product of two Gaussians $\mathcal{N}(\mathbf{x};\, \boldsymbol{\mu}_\mathsf{x},\, \boldsymbol{\Sigma}_\mathsf{x})$ and $\mathcal{N}(\mathbf{n};\, \boldsymbol{\mu}_\mathsf{x},\, \boldsymbol{\Sigma}_\mathsf{x})$, with variables which are the speech $\mathbf{x}$ and noise $\mathbf{n}$ as functions of $\mathbf{u}$. Any multivariate Gaussian can be decomposed into Gaussians for each dimension conditional on the previous dimensions. For example, for the speech,

$$\mathcal{N}(\mathbf{x};\, \boldsymbol{\mu}_\mathsf{x},\, \boldsymbol{\Sigma}_\mathsf{x}) = \mathcal{N}(x_1;\, \mu_{\mathsf{x},1},\, \sigma^2_{\mathsf{x},1})\mathcal{N}(x_2;\, \mu_{\mathsf{x},2|1},\, \sigma^2_{\mathsf{x},2|1}) \cdots \mathcal{N}(x_d;\, \mu_{\mathsf{x},d|1:d-1},\, \sigma^2_{\mathsf{x},d|1:d-1}), \tag{42}$$

where $\mu_{\mathsf{x},i|1:i-1}$ and $\sigma^2_{\mathsf{x},i|1:i-1}$ are functions of the previous dimensions $\mathbf{x}_{1:i-1}$.

Though their inputs are non-linear functions of $\mathbf{u}$, both Gaussians in $\gamma(\mathbf{u}; \boldsymbol{\alpha}, \mathbf{y}_t)$ can be factorised in this fashion. The integrand can then be factorised as

$$\gamma(\mathbf{u}; \boldsymbol{\alpha}, \mathbf{y}_t) = \gamma_1(u_1; \alpha_1, \mathbf{y}_t)\gamma_2(u_2; u_1, \boldsymbol{\alpha}_{1:2}, \mathbf{y}_t)\gamma_3(u_3; \mathbf{u}_{1:2}, \boldsymbol{\alpha}_{1:3}, \mathbf{y}_t) \cdots \gamma_d(u_d; \mathbf{u}_{1:d-1}, \boldsymbol{\alpha}_{1:d}, \mathbf{y}_t), \tag{43a}$$

with each factor defined as

$$\gamma_i(u_i; \mathbf{u}_{1:i-1}, \boldsymbol{\alpha}_{1:i}, \mathbf{y}_t) \triangleq \mathcal{N}\Big(x(u_i, \alpha_i, y_{t,i});\, \mu_{\mathsf{x},i|1:i-1},\, \sigma^2_{\mathsf{x},i|1:i-1}\Big)\mathcal{N}\Big(n(u_i, \alpha_i, y_{t,i});\, \mu_{\mathsf{n},i|1:i-1},\, \sigma^2_{\mathsf{n},i|1:i-1}\Big), \tag{43b}$$

where the means and variances are functions of the previous dimensions, as in (42).

The shape of the density $\gamma(u_i; \mathbf{u}^{(l)}_{1:i-1}, \boldsymbol{\alpha}^{(l)}_{1:i}, \mathbf{y}_t)$ thus depends on the current partial sample $(\mathbf{u}^{(l)}_{1:i-1}, \boldsymbol{\alpha}^{(l)}_{1:i})$. For each partial sample, therefore, a separate proposal distribution $\rho(u_i|\mathbf{u}^{(l)}_{1:i-1}, \boldsymbol{\alpha}^{(l)}_{1:i})$ must be found. Since the factors have the form of density as the one-dimensional γ in the previous section, the proposal distribution discussed there can be used.

The integral $\iint \gamma(\mathbf{u}, \boldsymbol{\alpha}; \mathbf{y}_t)\, d\boldsymbol{\alpha}\, d\mathbf{u}$, the value of interest, is the normalisation constant of $\gamma(\mathbf{u}, \boldsymbol{\alpha}; \mathbf{y}_t)$, which will again be called Z. To find Z by stepping through dimensions, it can be expressed as a sequence of incremental normalisation constants Z_i/Z_{i-1}. Given a sample set $\{(\mathbf{u}^{(l)}_{1:i-1}, \boldsymbol{\alpha}^{(l)}_{1:i-1})\}$, the approximation of the incremental normalisation constant is (when re-sampling is used)

$$\frac{\widetilde{Z_i}}{Z_{i-1}} = \frac{1}{L}\sum_{l=1}^{L} \frac{\gamma_i(\alpha^{(l)}_i; \mathbf{y}_t)}{\rho_i(\alpha^{(l)}_i)} \frac{\gamma_i(u^{(l)}_i; \mathbf{u}^{(l)}_{1:i-1}, \boldsymbol{\alpha}^{(l)}_{1:i}, \mathbf{y}_t)}{\rho_i(u^{(l)}_i|\mathbf{u}^{(l)}_{1:i-1}, \boldsymbol{\alpha}^{(l)}_{1:i})}, \tag{44}$$

where samples $\alpha^{(l)}_i$ are drawn from proposal distribution $\rho_i(\alpha_i)$, and samples $u^{(l)}_i$ are drawn from the appropriate $\rho_i(u_i|\mathbf{u}^{(l)}_{1:i-1}, \boldsymbol{\alpha}^{(l)}_{1:i})$. Like in the one-dimensional case, the proposal distribution $\rho_i(\alpha_i)$ can be set to the target distribution $p(\alpha_i)$, which is assumed independent for every i.

Van Dalen (2011, 7.3.2.4) gives details about an efficient method to compute the parameters of the Gaussians in (43b). At each dimension i, the one-dimensional Gaussian for the next dimension given each of the partial samples must be computed, which takes $O(i^2)$ time. The overall complexity for approximating the likelihood for one corrupted-speech vector is therefore $O(d^3 \cdot L)$.

Resampling duplicates higher-weight samples from the sample set and removes lower-weight ones between every dimension. Conceptually, it focuses effort on higher-probability regions. A sample drawn where the proposal distribution is greater than the integrand is likely to be removed by the resampling process. Where the proposal distribution under-covers, a sample may be assigned an arbitrarily high weight. Resampling can then result in a sparse sample set. In the extreme case, the resulting sample set consists of only copies of that one sample. This is another reason why (as discussed in the section *Importance sampling from the integrand*) it is more important that the proposal distribution has mass everywhere the integrand does than vice versa.

Because resampling can induce sparsity, the order in which the dimensions are traversed becomes important. The longer ago samples for one dimensions were drawn, the more likely they are to have duplicate entries for that dimension. The sample set will therefore be less varied for earlier dimensions. u_i and u_j are less dependent when $j - i$ is greater. In this work, the order of the dimensions is therefore $\alpha_1, u_1, \alpha_2, u_2, \dots \alpha_d, u_d$. If $\alpha^{(l)}_1, \dots, \alpha^{(l)}_d$ were drawn before $u^{(l)}_1, \dots, u^{(l)}_d$, then the set of samples $\alpha^{(l)}_1, \dots, \alpha^{(l)}_d$ would become considerably less varied in $i - 1$ rounds of

resampling. For higher i, $u_i^{(l)}$ would then be drawn with only a few or one unique $\alpha_i^{(l)}$, which would limit the accuracy of the approximation of the normalisation constant.

This section has discussed a transformation of the integral that gives the corrupted speech likelihood $p(\mathbf{y}_t)$ and how to apply sequential importance sampling to approximate the integral. The estimate from the sampling scheme is consistent, but not unbiased. This means that for a small sample cloud, the approximated value for $p(\mathbf{y}_t)$ may be generally overestimated or underestimated. However, as the sample cloud size increases, the resulting value converges to the real likelihood.

5. Distance to the actual distribution

Though it is standard practice in speech recognition research to judge speech recognition methods by word error rates, the sampling method proposed in section 4 is so computationally expensive that it is infeasible to use it in a speech recogniser. However, it is possible to assess compensation methods on the criterion that they aim to optimise: the closeness to the predicted distribution. This predicted distribution is formed by combining the clean speech HMM and the noise model with the mismatch function. This section will propose a method to compare compensation methods based on the KL divergence of the compensated system from the predicted distribution. Note that unlike van Dalen and Gales (2010), which found the KL divergence for one component, this paper finds the KL divergence over a complete speech recogniser.

The Kullback-Leibler (KL) divergence measures how far a distribution q is from a distribution p. The KL divergence is always non-negative; it is 0 if and only if the distributions are the same. It has therefore been used to find how well compensated components match an idealised distribution (e.g. Gales, 1995). In this work, the idealised distribution is the exact corrupted speech distribution. However, it would be more valuable to compare a whole hidden Markov model with its ideal version. An HMM is a distribution over state sequences Θ and observation sequences $\mathbf{Y}$.[2] The predicted distribution is the HMM that results from combining a clean speech HMM with independent and identically distributed noise. Model compensation methods approximate the predicted distribution with another HMM. Because the prior over the state sequences is the same for both distributions, the KL divergence between the predicted HMM p and its approximation q becomes equal to the weighted average of the per-state KL divergence (van Dalen, 2011, 6.1.1):

$$\mathcal{KL}(p\|q) = \sum_{\theta} p(\theta)\mathcal{KL}(p^{(\theta)}\|q^{(\theta)}), \qquad \mathcal{KL}(p^{(\theta)}\|q^{(\theta)}) \triangleq \int p^{(\theta)}(\mathbf{y}) \log \frac{p^{(\theta)}(\mathbf{y})}{q^{(\theta)}(\mathbf{y})}\, d\mathbf{y} \tag{45}$$

where $p(\theta)$ is the prior of state θ, which can be found from the training data occupancies, $p^{(\theta)}(\mathbf{y})$ is the predicted distribution for state θ, and $q^{(\theta)}(\mathbf{y})$ its approximation. In this paper, $q^{(\theta)}$ is the compensated distribution found for example with VTS, DPMC, or transformed-space sampling. The problem in computing this divergence is the one that motivates this whole paper: the predicted corrupted-speech distribution $p^{(\theta)}$ has no closed form. The two occurances of $p(\theta)$ in $\mathcal{KL}(p^{(\theta)}\|q^{(\theta)})$ will be handled separately.

The first problem is the term $p^{(\theta)}(\mathbf{y})$ inside the logarithm in (45). The KL divergence can be decomposed as

$$\mathcal{KL}(p^{(\theta)}\big\|q^{(\theta)}) = \mathcal{H}(p^{(\theta)}\big\|q^{(\theta)}) - \mathcal{H}(p^{(\theta)}), \tag{46a}$$

where the cross-entropy of p and q, and the entropy of p, are defined as

$$\mathcal{H}(p^{(\theta)}\big\|q^{(\theta)}) = -\int p^{(\theta)}(\mathbf{y}) \log q^{(\theta)}(\mathbf{y})\, d\mathbf{y}; \qquad \mathcal{H}(p^{(\theta)}) = -\int p^{(\theta)}(\mathbf{y}) \log p^{(\theta)}(\mathbf{y})\, d\mathbf{y}. \tag{46b}$$

The key insight is that the entropy depends only on $p^{(\theta)}$ (the predicted distribution for one state) and not on $q^{(\theta)}$. When comparing different approximations $q^{(\theta)}$ against a fixed $p^{(\theta)}$, the entropy is therefore constant and only the cross-entropy varies. Since the cross-entropy $\mathcal{H}(p^{(\theta)}\big\|q^{(\theta)})$ is then equal to the KL divergence up to a constant, it

[2] Note that whereas Hershey and Olsen (2007) views HMMs as distributions over observations, here they are distributions over the state sequence as well. This allows the KL divergence to implicitly take the word sequence into account.

gives the relative positions of different compensation methods. This does not, however, give an absolute divergence. When $q^{(\theta)}$ becomes equal to $p^{(\theta)}$, the cross-entropy becomes equal to the entropy, but the latter cannot be computed for the noise-corrupted speech distribution.[3] However, when $q^{(\theta)}$ is computed with the transformed-space sampling method from section 4.2, the cross-entropy converges to the entropy as the number of samples goes to infinity.

The second problem is that the cross-entropy $\mathcal{H}(p^{(\theta)} \| q^{(\theta)})$ still contains a term $p^{(\theta)}(\mathbf{y})$ outside the log. However, it is straightforward to draw samples from p. First the state identity $\theta^{(l)}$ is drawn from the state prior $p(\theta)$. Then, an observation is drawn from the mixture of Gaussians associated with the state, as described in section 2.2. Using L joint samples $(\theta^{(l)}, \mathbf{y}^{(l)})$, the average cross-entropy between p and q can be approximated:

$$\sum_{\theta} p(\theta)\mathcal{H}(p^{(\theta)} \| q^{(\theta)}) = -\sum_{\theta} p(\theta) \int p^{(\theta)}(\mathbf{y}) \log q^{(\theta)}(\mathbf{y})\, d\mathbf{y} \simeq -\frac{1}{L} \sum_{l} \log q^{(\theta)}(\mathbf{y}^{(l)}), \quad \theta^{(l)} \sim p(\theta), \quad \mathbf{y}^{(l)} \sim p^{(\theta)}(\mathbf{y}). \quad (47)$$

This approximation has one caveat: the distribution q is assumed to be normalised, and if not, then the result is not valid. This means that the Algonquin approximation, which does not yield a normalised distribution over $\mathbf{y}$, cannot be assessed in this way. As pointed out in section 4.2, the likelihood approximation of transformed-space sampling is biased, but consistent. This means that as the size of its sample cloud increases, q converges to being normalised.

This approximation to the cross-entropy can be used to compare model compensation methods, each yielding a different distribution q, by evaluating (47). A clean speech HMM with states θ, a noise model, and a mismatch function are required. These define the predicted corrupted speech distribution, also an HMM with states θ, but with the clean speech output corrupted as in section 2.2. First, joint samples $(\theta^{(l)}, \mathbf{y}^{(l)})$ are drawn from the predicted distribution p. Then, for each compensation method the approximate average cross-entropy to the predicted distribution is computed with (47). For compensation methods that pre-compute a corrupted-speech distribution, evaluating the compensated likelihood $q^{(\theta)}(\mathbf{y}^{(l)})$ means simply evaluating the pre-computed distribution for $\theta^{(l)}$ at $\mathbf{y}^{(l)}$. For non-parametric methods, on the other hand, evaluating $q^{(\theta)}(\mathbf{y}^{(l)})$ requires extensive computation for every sample. When $q^{(\theta)}$ is the transformed-space sampling method from section 4.2, another level of sampling takes place inside the evaluation of $q^{(\theta)}(\mathbf{y}^{(l)})$.

As the size of the sample cloud goes to infinity, the approximation to the likelihood that transformed-space sampling computes converges to the predicted likelihood. The cross-entropy $\mathcal{H}(p^{(\theta)} \| q^{(\theta)})$ therefore converges to the entropy. Though the KL divergence cannot be computed exactly because the entropy cannot be computed, the cross-entropy that the transformed-space sampling method converges to indicates the point where the KL divergence is 0.

6. Experiments

This work has aimed to find an accurate approximation to the corrupted speech likelihood. To assess the performance, this work will initially consider the cross-entropy for a range of compensation methods to the predicted distribution. This allows a detailed assessment of the performance of specific approximations. This chapter will consider the effects of parameterisations of the noise-corrupted speech distributions. These include assuming the corrupted speech Gaussian-distributed for one clean speech Gaussian, and a common approximation to the mismatch function: assuming the phase factor α to be fixed. Van Dalen (2011, 8.2.3) additionally considers the effect of diagonalising the covariance matrix of this Gaussian.

6.1. Set-up

This work uses a noise-corrupted version of the Resource Management (Price et al., 1988) task both for evaluation of the cross-entropy and the word error rate. The Resource Management task is a medium-vocabulary task, with a 1000-word vocabulary. Operations Room noise from the NOISEX-92 database (Varga and Steeneken, 1993) is added artificially with the noise scaled to yield SNRs of 20 dB, 14 dB, and 8 dB. The clean training data contains 109 speakers reading 3990 sentences, 3.8 hours of data. State-clustered cross-word triphone models with 6 components per mixture are built using the HTK RM recipe (Young et al., 2006). All word error rates are averaged over three of the four available test sets, Feb89, Oct89, and Feb91, a total of 30 test speakers and 900 utterances. The Python source

[3] The entropy could be rewritten to $\mathcal{H}(p) = \int p^{(\theta)}(\mathbf{y}) \log p^{(\theta)}(\mathbf{y})\, d\mathbf{y} = \int \left(\iint p(\mathbf{x}, \mathbf{n}, \mathbf{y}|\theta)\, d\mathbf{n}\, d\mathbf{x} \right) \log \left(\iint p(\mathbf{x}, \mathbf{n}, \mathbf{y}|\theta)\, d\mathbf{n}\, d\mathbf{x} \right) d\mathbf{y}$, which has no obvious closed form.

code for the software used for the cross-entropy experiments is available at `http://mi.eng.cam.ac.uk/~rcv25/cross-entropy/`.

As discussed in the introduction, in practice methods for noise robustness often estimate the noise model on test data in an unsupervised fashion. This work examines how accurate model compenstaion can be given the correct noise model, without considering how to estimate it. To eliminate the influence of the noise estimation algorithm, the noise is trained directly on the noise audio. For the cross-entropy experiments, the full-covariance noise and speech Gaussians are both over 24 log-spectral coefficients, at the equivalent of 0 dB.

The speech distribution is taken from the trained Resource Management system, single-pass retrained to yield a model in the log-spectral domain. To compensate one state, DPMC and VTS compensation operate per Gaussian; IDPMC on the whole mixture. The likelihood of one sample given the state identity is, obviously, the sum of component likelihoods weighted by the component weights. For model compensation methods, the component likelihoods follow straightforwardly from the pre-computed Gaussians. However, for transformed-space sampling, computing the component likelihoods requires extensive computation for each combination of component and observation. The state likelihoods for 8192 samples $\mathbf{y}^{(l)}$ from the corrupted speech distribution are computed, at which point the cross-entropy has converged.

The speech recogniser experiments compare various model compensation methods. Previous work had not applied model compensation with a phase factor distribution. Here, the experiments for both VTS and DPMC use a distribution over the phase factor α of the mismatch function (section 6.2 examines the influence of the phase factor).

An issue particular to model compensation is the treatment of *dynamic* parameters. Speech recognition systems add dynamic coefficients to the static coefficients. These approximate the change over time of the statics with a linear transformation of a window of consecutive static feature vectors. Thus if the window is ±1,

$$\mathbf{y}_t^{\Delta} = \begin{bmatrix} -\mathbf{I} & \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{y}_{t-1} \\ \mathbf{y}_t \\ \mathbf{y}_{t+1} \end{bmatrix} \simeq \left. \frac{\partial \mathbf{y}}{\partial t} \right|_{\mathbf{y}_t} . \tag{48}$$

Traditionally, model compensation methods use the approximation on the right-hand side. This is called the *continuous time approximation* (Gopinath et al., 1995). However, better performance is possible by modelling the distribution of the static vectors in a window directly. The vector containing the static vectors in a window is called the *extended* feature vector (van Dalen and Gales, 2011). This keeps the form of compensation close to that of the statics. For speech recognition, this paper will apply model compensation with extended feature vectors.

The number of samples for training DPMC is increased until they have converged. For the cross-entropy experiments, the number of samples per state is 720 000. The number of samples per component for extended DPMC is set to 100 000. For extended iterative DPMC (see van Dalen, 2011, 5.3.2), the average number of samples is 100 000: for example, a 6-component mixture is trained on 600 000 samples.

6.2. Results

If the speech and noise models represented the real distributions perfectly, then computing the corrupted speech distribution exactly would yield the best recognition performance. In practice, however, the models are imperfect and improving the KL divergence to the real distribution does not necessarily mean that the speech recognition accuracy will also improve. In this respect, assessing the quality of speech recognition compensation with the KL divergence is conceptually similar to assessing language models by their perplexities. The following sections will also relate cross-entropy results to word error rates.

However, not all methods discussed in this work can be assessed with both of these metrics. The Algonquin algorithm, discussed in section 3.3, yields a Gaussian approximation of the corrupted speech distribution specific to an observation. Used as a method to approximate the likelihood of observations, it therefore is not normalised. This makes it impossible to compute the cross-entropy for it. As discussed in section 4.2, the likelihood for transformed-space sampling is not normalised for small sample clouds, but converges to normalisation as the number of samples increases.

This work will not present word error rates for the transformed-space sampling method introduced in this work, because decoding with it is prohibitively slow. This is caused by the conceptual difference between model compensation methods (e.g., VTS, DPMC, and IDPMC) on the one hand and transformed-space sampling on the other. Model

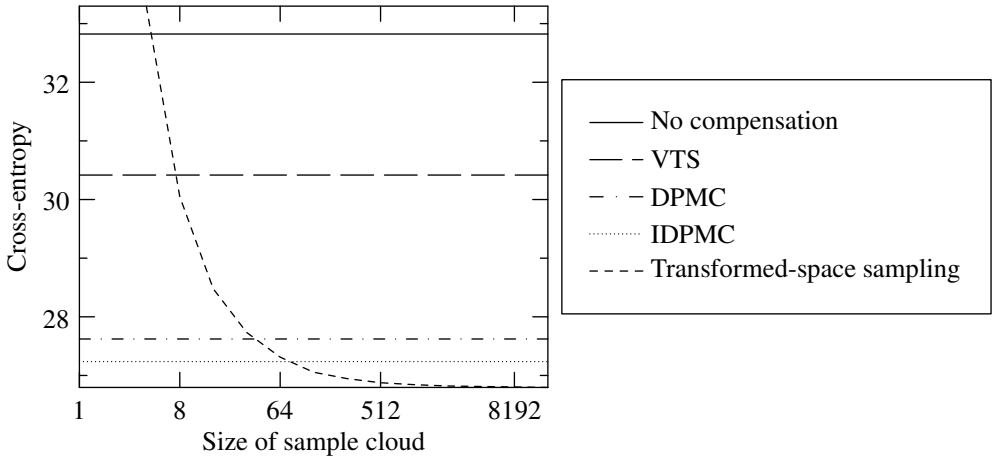


Figure 9: Cross-entropy to the corrupted speech distribution for transformed-space sampling and model compensation methods.

compensation computes a parametric distribution offline, and once that is done, running a recogniser or computing a cross-entropy is no slower than without compensation. Transformed-space sampling, on the other hand, has to work online, since it approximates the likelihood given an observation, and it cannot pre-compute anything. For a decent-sized sample cloud of 512, computing the likelihood of one sample for one speech Gaussian takes slightly over 30 seconds on a modern machine. For the recognition experiments in this work, just for the statics, it would run at roughly 30 million times real-time. This figure is based on an implementation that was not optimised for speed, but even with an optimised implementation running a speech recogniser with it would not be feasible. This is in contrast to the methods in Faubel and Wölfel (2007); Hershey et al. (2010). Both of these ignore covariances between features and use fewer Gaussians for the speech model. Faubel and Wölfel (2007) additionally uses feature enhancement, applying importance sampling to a different problem.

Compensation methods

The graph in figure 9 shows the cross-entropy from various compensated HMMs to the real predicted distribution. The curved line indicates the cross-entropy between the real distribution and the transformed-space sampling method described in section 4.2, for increasing sample cloud size.

As the size of the sample cloud increases, the approximation of $p(\mathbf{y}^{(l)})$ found with transformed-space sampling converges to the correct value. This means that the cross-entropy $\mathcal{H}(p\|q)$ converges to the entropy $\mathcal{H}(p)$. The bottom of the graph is set to the point the curve in figure 9 converges to, which indicates the entropy of p. Since the KL divergence can be written (in (46a)) as $\mathcal{KL}(p\|q) = \mathcal{H}(p\|q) - \mathcal{H}(p)$, this is the point where the KL divergence is 0. Since the KL divergence cannot be negative, this point gives the optimum cross-entropy. It gives a lower bound on how well the real corrupted speech distribution can be matched. The value of the cross-entropy for transformed-space sampling with 16 384 samples, 26.79, will also be the bottom for the other graph in this section.

The line labelled "DPMC" in figure 9 indicates the best match to the real distribution possible mapping one Gaussian onto one Gaussian. The Monte Carlo approximation to the cross-entropy, section 5 has shown, is equivalent to the negative average log-likelihood on the samples. DPMC finds the Gaussian that maximises its log-likelihood on the samples it is trained on. If the sample sets for training and testing were the same, then DPMC would yield the mathematically optimal Gaussian. Though different sample sets are used (with about 120 000 samples per Gaussian for training DPMC and 8192 in total samples for testing) the cross-entropy has converged. Any other Gaussian approximation would perform worse.

The state-of-the-art VTS compensation finds such a Gaussian analytically, and it is much faster. However, its cross-entropy to the real distribution is far from DPMC's ideal one.

Just like DPMC, IDPMC finds a distribution from samples, but it compensates a state-conditional mixture of

Compensation	Covariance	20 dB	14 dB	8 dB
—	diag	38.1	83.8	99.5
VTS, $\alpha = 1$		8.6	17.3	35.4
eVTS	full	11.1	16.5	28.2
eDPMC		7.4	13.3	27.4
eIDPMC		6.9	12.0	25.4
eIDPMC + 6		6.2	11.1	24.2
eIDPMC + 12		6.5	11.3	24.3

Table 1: Word error rates for various compensation schemes.

Gaussians at once rather than one Gaussian component. The mixture in the graph has 18 components trained on 720 000 samples, and comes closest of the parametric methods to the exact distribution. With an infinite number of components, it would yield the exact distribution. To correctly model the non-Gaussianity in 24 dimensions, however, a large number of components M are necessary. As section 3.1 has explained, the effective time complexity of IDPMC is $O(M^3)$, which quickly becomes infeasible.

To examine the link between the cross-entropy and the word error rate, recognition experiments are run. Improved modelling of the corrupted speech does not guarantee better discrimination, since speech and noise models are not necessarily the real ones.

Table 1 contains word error rates at two signal-to-noise ratios for comparison with the cross-entropy results in figure 9. Results with the uncompensated system, trained on clean data, are in the top row. Below it, as a reference, is standard VTS. The phase factor α is set to 1, the value that optimises the word error rate for this task and the noise model, and the compensated covariance matrices diagonal. Standard VTS uses the continuous time approximation to compensate delta- and delta-delta parameters. This yields inaccurate compensation for off-diagonals. This is discussed in great depth in van Dalen and Gales (2011). Using block-diagonal statistics and compensation (not in the table), word error rates for standard VTS are worse: 19.5 % and 38.5 %.

The bottom part of the table contains results on extended VTS (eVTS) and extended DPMC (eDPMC). They use a distribution over the phase factor α. The covariance matrices of the resulting distributions are full. As discussed in section 6.1, eVTS and eDPMC use distributions over extended feature vectors (van Dalen and Gales, 2011), which consist of the statics in a window that dynamic parameters are computed from. The original version of DPMC (Gales, 1995) assumes that the dynamic parameters are extracted using simple differences, so it could not be run without degrading the baseline.

eVTS performs less well than standard VTS at 20 dB. This is caused by the interaction of the phase factor with the vector Taylor series approximation, which section 6.2 will explore in more detail. At 14 dB, the more precise modelling does pay off. Compared to the uncompensated system extended VTS's performance improves more (38.1 % to 11.1 %) than expected from its improvement in terms of the cross-entropy in figure 9. VTS compensation uses a vector Taylor series approximation around the speech and noise means. It therefore models the mode of the corrupted speech distribution better than the tails. This causes the majority of the improvement in discrimination.

However, extended DPMC, which finds the optimal Gaussian given the speech and noise models, does yield better accuracy (7.4 %). Extended DPMC finds one corrupted speech Gaussian for one clean speech Gaussian. The cross-entropy experiment only used one clean speech Gaussian. Extended IDPMC (eIDPMC), however, trains a mixture of Gaussians from samples, which can be drawn from any distribution. For the recognition experiments, therefore, eIDPMC compensates one state-conditional mixture at a time. Replacing the 6-component speech distribution by a 6-component corrupted speech distribution, eIDPMC increases performance from eDPMC's 7.4 % to 6.9 %. By modelling the the distribution better, with 12 components ("eIDPMC + 6"), performance increases further to 6.2 %. The corrupted speech distribution should be more precise as the number of components increases to 18 ("eIDPMC + 12"). However, even by increasing the number of samples by a factor of 2, to 3600 000, performance does not increase. This can be explained by lack of robustness of the speech statistics, even though they have striped covariance matrices. This should be the best possible word error rate for these clean speech and noise distributions and this noise model.

Going from a Gaussian trained with extended VTS to the optimal Gaussian to a mixture of Gaussians in general improves the precision of the corrupted speech model. This shows in the cross-entropy to the real distribution, and the

same effects are observed in the word error rate. Better modelling of the corrupted speech distribution leads to better performance. The next sections will evaluate a specific approximation: setting α to a fixed value.

Influence of the phase factor

Model compensation often assumes a mismatch function that is an approximation to the real one as presented in section 2.1. However, traditionally the phase factor α, which arises from the interaction between the speech and the noise in the complex plane, has been assumed fixed. This section will look into the effect of the approximation of assuming the phase factor fixed.

Both for the predicted distribution and for the compensation methods, the phase factor distribution is assumed a zero-mean Gaussian, constrained to $[-1, +1]$. (For VTS compensation, the constraint to $[-1, +1]$ cannot be applied.) Previous work has not used DPMC with a phase factor distribution. VTS for feature enhancement has previously used such a distribution (Deng et al., 2004; Leutnant and Haeb-Umbach, 2009a). For model compensation, α has been set to fixed values (Li et al., 2007; Liao, 2007), but not to a distribution. Two settings for α are of interest. The traditional presentation (Moreno, 1996; Acero et al., 2000) sets $\alpha = 0$, which is the mean of the actual distribution. The second setting is $\alpha = 1$, which is roughly equivalent to ignoring the term with α in the mismatch function and using magnitude-spectrum feature vectors (van Dalen, 2011, appendix C.2). This has been applied in previous work (e.g. Liao, 2007).

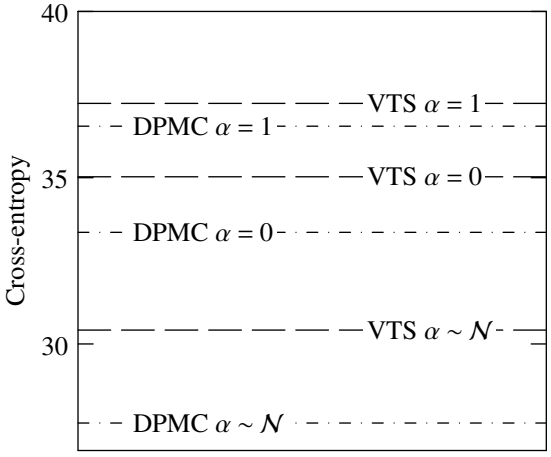


Figure 10: The effect of the phase factor on Gaussian compensation.

Figure 10 shows the cross-entropy for DPMC and VTS with different models for the phase factor. Note that the vertical axis uses a larger scale than figure 9. The bottom of the graph is still set to the optimal cross-entropy acquired with transformed-space sampling. Both methods generate full covariance matrices. The diagonalised versions (not shown in the table) show the same trends, with smaller distances between cross-entropies. DPMC with the model for α matching the actual distribution ("DPMC $\alpha \sim \mathcal{N}$") yields the lowest cross-entropy by definition. VTS with a Gaussian model ("VTS $\alpha \sim \mathcal{N}$") is at some distance.

The obvious choice for fixing α is the mean of its actual distribution, 0. With that assumption, both DPMC and VTS end up further away from the ideal distribution. As expected, when α is fixed to 1, the modelled distributions become even further away from the actual ones.

Table 2 contains word error rates for the same contrasts. Compensation with extended feature vectors is used. The form of compensation of the dynamic parameters is therefore more closely related to that of the static parameters. Additionally, it allows full-covariance compensation, word error rates for which are in the table. With diagonal covariances, the trends are again the same but less pronounced. For eDPMC, the effect of different phase factor models is as expected. Whether α is distributed around 0 or fixed to 0 mostly affects the covariances. Though this does have an effect on the cross-entropy, this makes little difference for discrimination, since the change to the covariance matrices is fairly uniform across components. However, setting α to the inconsistent value 1 affects performance negatively.

Scheme	α	20 dB	14 dB
eDPMC	0	7.6	13.2
	1	8.0	14.7
	$\mathcal{N}$	7.4	13.3
eVTS	0	11.4	16.5
	1	8.7	14.9
	$\mathcal{N}$	11.1	16.5

Table 2: The effect of the phase factor on Gaussian compensation: word error rates.

The results for VTS are more surprising. Again, there is little difference between setting α to 0 and letting it be distributed around 0. For VTS, this by definition does not affect components' means, but only their covariances. However, setting it to 1 does improve performance. This may be because overestimation of the mode (see section 3.2) improves modelling for some components. This suggests that for different tasks, different settings for α optimise compensation for components where mis-compensation is most likely to cause recognition errors. This would explain why the optimal α is different for different tasks (Gales and Flego, 2010).

What the results here do show is that while modelling α with a distribution reduces the distance to the actual distribution, as evidenced by the improving cross-entropy, discrimination is not helped. Section 3.2 has pointed out that the only effect of using a distribution for the phase factor over a fixed value at the distribution's mode is a fairly equal bias on the covariance, which is almost equal for adjacent components, and is therefore unlikely to influence discrimination much. It has also discussed how in practice the noise estimation can subsume this bias. Using a distribution over the phase factor rather than a fixed value, as is currently done, is therefore unlikely to cause gains in a practical speech recogniser.

7. Conclusion

This work has introduced a new technique for computing the likelihood of a corrupted speech observation vector. Instead of a parametric density, it applies a sampling method, which approximates the integral over speech, noise and phase factor that the likelihood consists of. Because the integrand has an awkward shape, it is first transformed. Then, sequential importance re-sampling deals with the high dimensionality. As the number of samples tends to infinity, and given models for the speech, the noise, and a mismatch function, this approximation converges to the real likelihood.

Because the method cannot pre-compute distributions, it is too slow to embed in a speech recogniser. However, it is possible to find the KL divergence from approximations to the corrupted speech distribution to the real one up to a constant. The new method essentially gives the point where the KL divergence is 0, so it can be assessed how close compensation methods are to the ideal. For the recognition experiments, the compensation schemes work on extended feature vectors to provide the best compensation and to avoid further approximations. This work has introduced model compensation using a phase factor distribution for eVTS and eDPMC. The KL divergence for different compensation methods generally appears to predict their word error rates. Standard VTS or extended VTS compensation with an optimised mismatch function result in word error rates close to that of the optimal Gaussian compensation. A compensation method that does not assume Gaussianity is iterative data-driven parallel model combination (IDPMC), which takes impractically long to train but it is feasible to run speech recognition with. In terms of the KL divergence, it slowly converges to the real corrupted-speech distribution. In the recognition experiments, it performs substantially better than the optimal Gaussian compensation. This demonstrates that more accurate model compensation than Gaussian-for-Gaussian improves the performance of speech recognition.

References

Acero, A., 1990. Acoustical and Environmental Robustness in Automatic Speech Recognition. Ph.D. thesis. Carnegie Mellon University.

Acero, A., Deng, L., Kristjansson, T., Zhang, J., 2000. HMM adaptation using vector Taylor series for noisy speech recognition, in: Proceedings of ICSLP, pp. 229–232.

Bishop, C.M., 2006. Pattern Recognition and Machine Learning. Springer.

van Dalen, R.C., 2011. Statistical Models for Noise-Robust Speech Recognition. Ph.D. thesis. University of Cambridge.

van Dalen, R.C., Gales, M.J.F., 2009. Extended VTS for noise-robust speech recognition. Technical Report CUED/F-INFENG/TR.636. Cambridge University Engineering Department.
van Dalen, R.C., Gales, M.J.F., 2010. Asymptotically exact noise-corrupted speech likelihoods, in: Proceedings of Interspeech, pp. 709–712.
van Dalen, R.C., Gales, M.J.F., 2011. Extended VTS for noise-robust speech recognition. IEEE Transactions on Audio, Speech, and Language Processing 19, 733–743.
Deng, L., Droppo, J., Acero, A., 2004. Enhancement of log mel power spectra of speech using a phase-sensitive model of the acoustic environment and sequential estimation of the corrupting noise. IEEE Transactions on Speech and Audio Processing 12, 133–143.
Doucet, A., Johansen, A.M., 2008. A tutorial on particle filtering and smoothing: fifteen years later. Technical Report. Department of Statistics, University of British Columbia.
Du, J., Huo, Q., 2008. A speech enhancement approach using piecewise linear approximation of an explicit model of environmental distortions, in: Proceedings of Interspeech, pp. 569–572.
Ephraim, Y., 1990. A mimimum mean square error approach for speech enhancement, in: Proceedings of ICASSP, pp. 829–832.
Faubel, F., Wölfel, M., 2007. Overcoming the VTS approximation in speech feature enhancement — a particle filter approach, in: Proceedings of ICASSP, pp. 557–560.
Flego, F., Gales, M.J.F., 2011. Factor Analysis Based VTS and JUD Noise Estimation and Compensation. Technical Report CUED/F-INFENG/TR.653. Cambridge University.
Frey, B.J., Deng, L., Acero, A., Kristjansson, T., 2001a. ALGONQUIN: Iterating Laplace's method to remove multiple types of acoustic distortion for robust speech recognition, in: Proceedings of Eurospeech, pp. 901–904.
Frey, B.J., Kristjansson, T.T., Deng, L., Acero, A., 2001b. ALGONQUIN: Learning dynamic noise models from noisy speech for robust speech recognition, in: Proceedings of NIPS.
Gales, M.J.F., 1995. Model-Based Techniques for Noise Robust Speech Recognition. Ph.D. thesis. Cambridge University.
Gales, M.J.F., van Dalen, R.C., 2007. Predictive linear transforms for noise robust speech recognition, in: Proceedings of ASRU, pp. 59–64.
Gales, M.J.F., Flego, F., 2010. Discriminative classifiers with adaptive kernels for noise robust speech recognition. Computer Speech and Language 24, 648–662.
Gopinath, R.A., Gales, M.J.F., Gopalakrishnan, P.S., Balakrishnan-Aiyer, S., Picheny, M.A., 1995. Robust speech recognition in noise - performance of the IBM continuous speech recognizer on the ARPA noise spoke task, in: Proceedings of the ARPA Workshop on Spoken Language System Technology, pp. 127–130.
Hershey, J.R., Olsen, P., Rennie, S.J., 2010. Signal interaction and the devil function, in: Proceedings of Interspeech, pp. 334–337.
Hershey, J.R., Olsen, P.A., 2007. Approximating the Kullback Leibler divergence between Gaussian mixture models, in: Proceedings of ICASSP.
Holmes, J., Sedgwick, N., 1986. Noise compensation for speech recognition using probabilistic models, in: Proceedings of ICASSP, pp. 741–744.
Julier, S.J., Uhlmann, J.K., 2004. Unscented filtering and nonlinear estimation. Proceedings of the IEEE 92, 401–422.
Kim, D., Un, C., Kim, N., 1998. Speech recognition in noisy environments using first-order vector Taylor series. Speech Communication 24, 39–49.
Kitagawa, G., 1996. Monte Carlo filter and smoother for non-Gaussian non-linear state space models. Journal of Computational and Graphical Statistics 5, 1–25.
Klatt, D., 1976. A digital filter bank for spectral matching, in: Proceedings of ICASSP, pp. 573–576.
Kristjansson, T., Frey, B., Deng, L., Acero, A., 2001. Joint estimation of noise and channel distortion in a generalized EM framework, in: Proceedings of ASRU.
Kristjansson, T.T., Frey, B.J., 2002. Accounting for uncertainity [*sic*] in observations: a new paradigm for robust automatic speech recognition, in: Proceedings of ICASSP, pp. 61–64.
Leutnant, V., Haeb-Umbach, R., 2009a. An analytic derivation of a phase-sensitive observation model for noise robust speech recognition, in: Proceedings of Interspeech, pp. 2395–2398.
Leutnant, V., Haeb-Umbach, R., 2009b. An analytic derivation of a phase-sensitive observation model for noise robust speech recognition. Technical Report. Universität Paderborn, Faculty of Electrical Engineering and Information Technology.
Li, J., Deng, L., Yu, D., Gong, Y., Acero, A., 2007. High-performance HMM adaptation with joint compensation of additive and convolutive distortions via vector Taylor series, in: Proceedings of ASRU, pp. 65–70.
Li, J., Yu, D., Deng, L., Gong, Y., Acero, A., 2009. A unified framework of HMM adaptation with joint compensation of additive and convolutive distortions. Computer Speech and Language 23, 389–405.
Li, J., Yu, D., Gong, Y., Deng, L., 2010. Unscented transform with online distortion estimation for HMM adaptation, in: Proceedings of Interspeech, pp. 1660–1663.
Liao, H., 2007. Uncertainty Decoding for Noise Robust Speech Recognition. Ph.D. thesis. Cambridge University.
Liao, H., Gales, M.J.F., 2005. Uncertainty decoding for noise robust speech recognition, in: Proceedings of Interspeech, pp. 3129–3132.
Moreno, P.J., 1996. Speech Recognition in Noisy Environments. Ph.D. thesis. Carnegie Mellon University.
Myrvoll, T.A., Nakamura, S., 2004. Minimum mean square error filtering of noisy cepstral coefficients with applications to ASR, in: ICASSP, pp. 977–980.
Price, P., Fisher, W.M., Bernstein, J., Pallett, D.S., 1988. The DARPA 1000-word Resource Management database for continuous speech recognition, in: Proceedings of ICASSP, pp. 651–654.
Sagayama, S., Yamaguchi, Y., Takahashi, S., Takahashi, J., 1997. Jacobian approach to fast acoustic model adaptation, in: Proceedings of ICASSP, pp. 835 – 838.
Seltzer, M., Acero, A., Kalgaonkar, K., 2010. Acoustic model adaptation via linear spline interpolation for robust speech recognition, in: Proceedings of ICASSP, pp. 4550–4553.
Shinohara, Y., Akamine, M., 2009. Bayesian feature enhancement using a mixture of unscented transformations for uncertainty decoding of noisy speech, in: Proceedings of ICASSP, pp. 4569–4572.
Stouten, V., 2006. Robust automatic speech recognition in time-varying environments. Ph.D. thesis. Katholieke Universiteit Leuven.
Stouten, V., Van hamme, H., Wambacq, P., 2005. Effect of phase-sensitive environment model and higher order VTS on noisy speech feature

enhancement, in: Proceedings of ICASSP, pp. 433–436.
Varga, A., Steeneken, H.J.M., 1993. Assessment for automatic speech recognition II: NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems. Speech Communication 12, 247–251.
Xu, H., Chin, K.K., 2009. Joint uncertainty decoding with the second order approximation for noise robust speech recognition, in: Proceedings of ICASSP, pp. 3841–3844.
Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Liu, X.A., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., Woodland, P., 2006. The HTK book (for HTK version 3.4).
Yu, D., Deng, L., Droppo, J., Wu, J., Gong, Y., Acero, A., 2008. Robust speech recognition using a cepstral minimum-mean-square-error-motivated noise suppressor. IEEE Transactions on Audio, Speech, and Language Processing 16, 1061–1070.