

Asymptotically Exact Noise-Corrupted Speech Likelihoods

R. C. van Dalen, M. J. F. Gales

Cambridge University Engineering Department, UK

rcv25@cam.ac.uk, mjfg@eng.cam.ac.uk

Abstract

Model compensation techniques for noise-robust speech recognition approximate the corrupted speech distribution. This paper introduces a sampling method that, given speech and noise distributions and a mismatch function, in the limit calculates the corrupted speech likelihood exactly. Though it is too slow to compensate a speech recognition system, it enables a more fine-grained assessment of compensation techniques, based on the KL divergence of individual components. This makes it possible to evaluate the impact of approximations that compensation schemes make, such as the form of the mismatch function.

Index Terms: speech recognition, noise robustness

1. Introduction

Background noise can severely impact the performance of speech recognisers. This degradation in performance is caused by the difference between training and test conditions. Model compensation methods aim to find a corrupted speech distribution appropriate to the test condition. They usually map single clean speech Gaussians to single corrupted speech Gaussians.

However, the corrupted speech distributions are not Gaussian: the likelihood expression does not even have a closed form. In contrast with model compensation schemes, this work introduces a sampling method that approximates the likelihood for a given observation vector. The integral in the likelihood expression is rewritten to allow importance sampling to be used. In the limit, this likelihood calculation is exact.

This new scheme is too slow to compensate a speech recogniser. However, it can be used to assess how well a compensation scheme matches the correct distribution, based on the KL divergence. The impact of various aspects of compensation schemes can thus be evaluated: assuming the corrupted speech Gaussian, assuming independence between dimensions, and ignoring the phase differences between speech and noise. The KL divergence is shown to predict speech recogniser accuracy.

In this work, the mismatch function and the distributions of the speech and additive noise are assumed to be known. Since the convolutional noise merely causes an offset on the feature vectors, it is not explicitly considered.

2. The mismatch function

The relationship between the corrupted speech \mathbf{y} , the clean speech \mathbf{x} , and the noise \mathbf{n} is central to noise-robust speech recognition. The theory and the cross-entropy experiments in this work use log-spectral features. In the log-power-spectral domain, this relationship is independent per dimension. This “mismatch function” is [1, 2]

$$\exp(\mathbf{y}) = \exp(\mathbf{x}) + \exp(\mathbf{n}) + 2\alpha \circ \exp\left(\frac{1}{2}\mathbf{x} + \frac{1}{2}\mathbf{n}\right), \quad (1)$$

Rogier van Dalen is sponsored by Toshiba Research Europe Ltd.

where $\exp(\cdot)$, $\log(\cdot)$, and \circ indicate element-wise exponentiation, logarithm, and multiplication, respectively. The distribution of \mathbf{y} depends on those of the speech \mathbf{x} and the noise \mathbf{n} , and the phase factor α .

This work uses the standard assumption that the noise is independent of the speech and Gaussian distributed. The speech distribution is usually taken from a recogniser trained on clean speech. This paper focuses on a single speech Gaussian; the dependence on the component is dropped for clarity. Thus, the distributions of the speech \mathbf{x} and the noise \mathbf{n} are

$$\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x); \quad \mathbf{n} \sim \mathcal{N}(\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n). \quad (2)$$

Since in the log-spectral domain dimensions are strongly correlated, the covariance matrices $\boldsymbol{\Sigma}_x$ and $\boldsymbol{\Sigma}_n$ are full.

The phase factor α in (1) arises from the interaction of the spectra of the speech and noise signals in the complex domain. α is often assumed equal to its expected value, $\mathbf{0}$ [3, 4]. Recently, however, interest has grown in incorporating a phase-sensitive mismatch function in methods for noise-robustness [1, 2] since this models reality more accurately. It can be shown that elements α_i of α are constrained to $[-1, 1]$ [1]. Their second moments ($\sigma_{\alpha_i}^2$) can be approximated well from just the shape of the corresponding filter bins [2]. Since they are roughly Gaussian distributed [5], this work approximates the distribution of α_i with

$$p(\alpha_i) \propto \begin{cases} \mathcal{N}(\alpha_i; 0, \sigma_{\alpha_i}^2) & \alpha_i \in [-1, +1]; \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

3. Parametric likelihood representations

Model compensation methods approximate the corrupted speech with a parametric distribution. Often a Gaussian is used, so that $\mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y)$. The following briefly discusses three existing compensation methods.

VTS compensation [6, 4, 1] is a standard approach that replaces the mismatch function with a first-order vector Taylor series approximation. Here, the standard form is generalised to include the phase factor α [1] for model compensation. The corrupted speech then becomes Gaussian with parameters

$$\boldsymbol{\mu}_y = \mathbf{f}(\boldsymbol{\mu}_x, \boldsymbol{\mu}_n, \boldsymbol{\mu}_\alpha); \quad \boldsymbol{\Sigma}_y = \mathbf{J}_x \boldsymbol{\Sigma}_x \mathbf{J}_x^\top + \mathbf{J}_n \boldsymbol{\Sigma}_n \mathbf{J}_n^\top + \mathbf{J}_\alpha \boldsymbol{\Sigma}_\alpha \mathbf{J}_\alpha^\top,$$

where $\mathbf{f}(\boldsymbol{\mu}_x, \boldsymbol{\mu}_n, \boldsymbol{\mu}_\alpha)$ is the observation vector obtained by setting the other variables to their means, and \mathbf{J}_x , \mathbf{J}_n , and \mathbf{J}_α are the partial derivatives of \mathbf{y} with respect to \mathbf{x} , \mathbf{n} , and α .

A more accurate but slower approach for finding the parameters of a Gaussian is data-driven parallel model combination (DPMC) [3]. This draws S samples $(\mathbf{x}^{(s)}, \mathbf{n}^{(s)}, \alpha^{(s)})$ for the speech, noise and phase factor and computes corrupted speech samples $\mathbf{y}^{(s)} = \mathbf{f}(\mathbf{x}^{(s)}, \mathbf{n}^{(s)}, \alpha^{(s)})$.¹ The parameters of the

¹This is a straightforward extension to the original DPMC algorithm, which does not use the phase factor α .

Gaussian are then set to their maximum-likelihood values:

$$\boldsymbol{\mu}_y = \frac{1}{S} \sum_{s=1}^S \mathbf{y}^{(s)}; \quad \boldsymbol{\Sigma}_y = \frac{1}{S} \sum_{s=1}^S \mathbf{y}^{(s)} \mathbf{y}^{(s)\top} - \boldsymbol{\mu}_y \boldsymbol{\mu}_y^{\top}.$$

In the limit as the number of samples S goes to infinity, this finds the optimal Gaussian parameters.

A third model compensation method, iterative DPMC [3], also trains the corrupted speech parameters on samples, but the distribution is a mixture of Gaussians, and training uses expectation–maximisation. It is possible to draw speech samples from a state-conditional distribution (usually a mixture of Gaussians) and train a mixture with a different number of components. By increasing the number of components, the corrupted speech distribution can be matched arbitrarily well. In practice, the number of iterations of EM needed increases linearly with the number of components, and to train the mixture well, the number of samples needs to increase more than linearly. The effective computational time therefore increases more than quadratically, so that IDPMC quickly becomes prohibitively slow.

4. Per-observation likelihood evaluation

The previous section has discussed parametric approximations to the corrupted speech distribution. However, no expression for the complete density is needed: while recognising speech only the likelihood of vectors that are observed is required. Therefore, in this section the likelihood is approximated for a specific observation \mathbf{y}_t . The exact likelihood is

$$p(\mathbf{y}_t) = \iint p(\mathbf{y}_t | \mathbf{x}, \mathbf{n}) p(\mathbf{n}) d\mathbf{n} p(\mathbf{x}) d\mathbf{x} \quad (4a)$$

$$= \iint \left[\int \delta_{\mathbf{f}(\mathbf{x}, \mathbf{n}, \boldsymbol{\alpha})}(\mathbf{y}_t) p(\boldsymbol{\alpha}) d\boldsymbol{\alpha} \right] p(\mathbf{n}) d\mathbf{n} p(\mathbf{x}) d\mathbf{x}, \quad (4b)$$

where $\delta_{\mathbf{f}(\mathbf{x}, \mathbf{n}, \boldsymbol{\alpha})}(\cdot)$ denotes a Dirac delta at the observation vector that results from specific \mathbf{x} , \mathbf{n} , and $\boldsymbol{\alpha}$.²

4.1. The Algonquin algorithm

It is possible to approximate the corrupted speech distribution with an observation-specific Gaussian. Like VTS compensation, the Algonquin algorithm [7] uses a first-order vector Taylor series approximation. The difference is that Algonquin iteratively updates the expansion point of the mismatch function. The influence of $\boldsymbol{\alpha}$ on \mathbf{y}_t is modelled zero-mean, fixed-covariance Gaussian. Linearising the influence of Gaussian distributed \mathbf{x} and \mathbf{n} on \mathbf{y}_t causes them to be jointly Gaussian. At iteration k ,

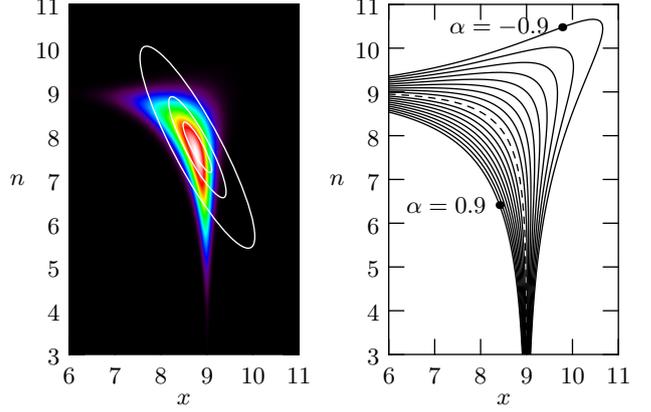
$$\begin{bmatrix} \mathbf{x} \\ \mathbf{n} \\ \mathbf{y}_t \end{bmatrix} \sim \mathcal{N} \left(\begin{bmatrix} \boldsymbol{\mu}_x \\ \boldsymbol{\mu}_n \\ \boldsymbol{\mu}_y \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_x & \mathbf{0} & \boldsymbol{\Sigma}_{xy}^{(k)} \\ \mathbf{0} & \boldsymbol{\Sigma}_n & \boldsymbol{\Sigma}_{ny}^{(k)} \\ \boldsymbol{\Sigma}_{yx}^{(k)} & \boldsymbol{\Sigma}_{yn}^{(k)} & \boldsymbol{\Sigma}_y^{(k)} \end{bmatrix} \right). \quad (5)$$

The Gaussian approximation for the distribution of \mathbf{y}_t is $q_{y_t}^{(k)}(\mathbf{y}_t) = \mathcal{N}(\boldsymbol{\mu}_y^{(k)}, \boldsymbol{\Sigma}_y^{(k)})$. Algonquin for model compensation effectively uses this distribution to compute the likelihood.³ However, since the parameters of $q_{y_t}^{(k)}(\mathbf{y}_t)$ depend on \mathbf{y}_t itself, it is not a normalised distribution over \mathbf{y}_t .

The joint Gaussian implies a Gaussian approximation to the posterior of \mathbf{x} and \mathbf{n} given \mathbf{y}_t , $q_{y_t}^{(k)}(\mathbf{x}, \mathbf{n})$. The expansion point

²Model compensation methods approximate this expression; indeed, [5] derives VTS and DPMC from (4b).

³This is multiplied by a per-frame normalisation term, but this does not affect decoding. [5] gives more details.



(a) $\gamma(x, n)$ for $x \sim \mathcal{N}(10, 1)$; $n \sim \mathcal{N}(9, 2)$; $\sigma_\alpha^2 = 0.04$. (b) x, n for various values of α .

Figure 1: (x, n) -space. $y_t = 9$.

for the next iteration, $k+1$ is set to the mean of this distribution. The next section attempts to draw samples from $q_{y_t}^{(k)}(\mathbf{x}, \mathbf{n})$ to use in importance sampling.

4.2. Integrating over \mathbf{x} and \mathbf{n}

This section introduces a method to approximate the integral in (4a) with Monte Carlo. It requires that $p(\mathbf{y}_t | \mathbf{x}, \mathbf{n})$ can be evaluated at any point (\mathbf{x}, \mathbf{n}) . Given \mathbf{x} and \mathbf{n} , \mathbf{y}_t and $\boldsymbol{\alpha}$ are deterministically related. Therefore, the space of the distribution in (4a) can be transformed from \mathbf{y}_t to $\boldsymbol{\alpha}$ by taking into account the Jacobian:

$$p(\mathbf{y}_t) = \iint \left| \frac{\partial \boldsymbol{\alpha}(\mathbf{x}, \mathbf{n}, \mathbf{y}_t)}{\partial \mathbf{y}} \right|_{\mathbf{y}_t} p(\boldsymbol{\alpha}(\mathbf{x}, \mathbf{n}, \mathbf{y}_t)) p(\mathbf{n}) p(\mathbf{x}) d\mathbf{n} d\mathbf{x} \\ \triangleq \iint \gamma(\mathbf{x}, \mathbf{n}) d\mathbf{n} d\mathbf{x}, \quad (6)$$

where $p(\boldsymbol{\alpha}(\mathbf{x}, \mathbf{n}, \mathbf{y}_t))$ denotes the density of $p(\boldsymbol{\alpha})$ at the value of $\boldsymbol{\alpha}$ implied by \mathbf{x} , \mathbf{n} , and \mathbf{y}_t . This expression is exact.⁴ Though the integrand γ is now straightforward to evaluate, the integral has no closed form. It can, however, be approximated with a Monte Carlo method. The interest here is in the integral rather than the samples, which rules out most Monte Carlo methods. *Importance sampling* does find the integral under a target density. It draws samples from a *proposal distribution* ρ and makes up for the difference between target and proposal densities with a weight factor $\gamma(\mathbf{x}, \mathbf{n})/\rho(\mathbf{x}, \mathbf{n})$. A good tutorial is [9].

The integral in (6) can be approximated with S weighted samples $(\mathbf{x}^{(s)}, \mathbf{n}^{(s)})$ from ρ :

$$\iint \frac{\gamma(\mathbf{x}, \mathbf{n})}{\rho(\mathbf{x}, \mathbf{n})} \rho(\mathbf{x}, \mathbf{n}) d\mathbf{n} d\mathbf{x} \simeq \sum_{s=1}^S \frac{\gamma(\mathbf{x}^{(s)}, \mathbf{n}^{(s)})}{\rho(\mathbf{x}^{(s)}, \mathbf{n}^{(s)})}. \quad (7)$$

An obvious choice for ρ is the Gaussian approximation of the posterior that the Algonquin algorithm finds. Figure 1a shows a one-dimensional example of $\gamma(\mathbf{x}, \mathbf{n})$, and, in white, the Gaussian approximation found with the Algonquin algorithm. Algonquin has placed the mode of the Gaussian on the actual mode. No Gaussian, however, can capture the curve in (\mathbf{x}, \mathbf{n}) -space. This results in two problems for importance sampling.

⁴For a more extensive derivation, see [5]. An approximation to (6) was given in [8], section 5.3.2.

Where the proposal distribution has a higher density than the target, samples are drawn almost uselessly. Where the proposal distribution is lower, few samples are drawn but they are assigned high weights. Many samples are then required for sufficient coverage. The number of samples required increases exponentially with dimensionality, so that it turns out infeasible to apply this scheme to a 24-dimensional log-spectral space.

4.3. Transformed-space integration

The problem with the scheme in section 4.2 is the hard-to-approximate bend in the distribution of \mathbf{x} and \mathbf{n} given an observation \mathbf{y}_t . The lines in Figure 1b indicate the possible values for x and n for observed $y_t = 9$ and different values of α . To deal with this bend, in this section the integration over \mathbf{x} and \mathbf{n} is replaced by an integration over a new variable \mathbf{u} . The two variable changes this requires are somewhat similar to the one in (6). [10] also replaced \mathbf{x} and \mathbf{n} by a new variable, but used a different substitution for the integral to be approximated with line segments, which meant log-spectral dimensions of the speech and noise had to be assumed independent.

Here, the substitute variable \mathbf{u} corresponds to a (\mathbf{x}, \mathbf{n}) -pair on a curve in Figure 1b. It relates \mathbf{x} and \mathbf{n} symmetrically: $\mathbf{u} = \mathbf{n} - \mathbf{x}$. A full derivation of the new integral is given in [5]. Here only the result is given. The two Jacobians resulting from the transformation of the space cancel out. The complete corrupted speech likelihood in (4b) becomes

$$p(\mathbf{y}_t) = \int p(\alpha) \int p(\mathbf{x}(\mathbf{u}, \alpha, \mathbf{y}_t)) p(\mathbf{n}(\mathbf{u}, \alpha, \mathbf{y}_t)) d\mathbf{u} d\alpha \\ \triangleq \int p(\alpha) \int \gamma(\mathbf{u}|\alpha) d\mathbf{u} d\alpha, \quad (8)$$

where $p(\mathbf{x}(\mathbf{u}, \alpha, \mathbf{y}_t))$ denotes the density of $p(\mathbf{x})$ at the value of \mathbf{x} implied by $\mathbf{u}, \alpha, \mathbf{y}_t$, and similar for $p(\mathbf{n}(\mathbf{u}, \alpha, \mathbf{y}_t))$. Again, this expression is exact, but has no closed form.

The factorisation makes it possible to draw samples $\alpha^{(s)}$ from $p(\alpha)$ and then use importance sampling for $\gamma(\mathbf{u}|\alpha^{(s)})$:

$$\int p(\alpha) \int \frac{\gamma(\mathbf{u}|\alpha)}{\rho(\mathbf{u}|\alpha)} \rho(\mathbf{u}|\alpha) d\mathbf{u} d\alpha \simeq \frac{1}{S} \sum_{s=1}^S \frac{\gamma(\mathbf{u}^{(s)}|\alpha^{(s)})}{\rho(\mathbf{u}^{(s)}|\alpha^{(s)})}, \quad (9)$$

where $\rho(\mathbf{u}|\alpha^{(s)})$ is a proposal distribution that approximates $\gamma(\mathbf{u}|\alpha^{(s)})$. However, the number of samples required still grows exponentially with the number of dimensions. To overcome this problem, *sequential importance resampling* [9] is applied. This keeps track of a cloud of samples that it extends one dimension at a time. Between dimensions, a process called *resampling* duplicates high-weight samples and removes low-weight samples. This focuses effort on the region of interest. To apply sequential importance sampling, both the integrand and the proposal distribution need to be factorised, which is not trivial with full-covariance speech and noise Gaussians. [5] details the factorisation and the form of proposal distributions.

The resulting estimate of the integral cannot be shown to be unbiased, but it is consistent: as the number of samples becomes higher, this approximation tends to the correct value of $p(\mathbf{y}_t)$. Apart from speed issues, this value could be used in a speech recogniser in place of the likelihood computation.

5. Distance to the actual distribution

The KL divergence could be used to assess how close model compensation methods are to the real distribution. If p is the

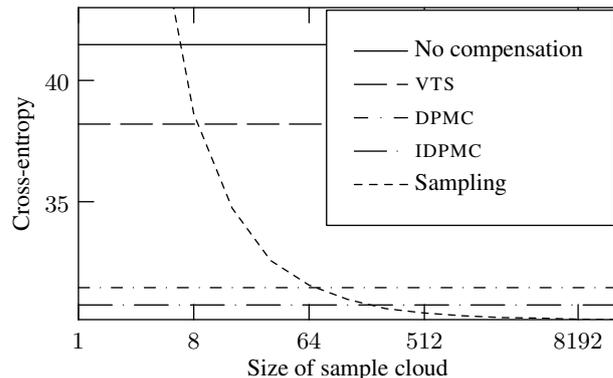


Figure 2: Cross-entropy to the corrupted speech distribution.

real distribution and q its approximation, the KL divergence is

$$\mathcal{KL}(p||q) = \int p(\mathbf{y}) \log \frac{p(\mathbf{y})}{q(\mathbf{y})} d\mathbf{y} = \mathcal{H}(p||q) - \mathcal{H}(p), \quad (10)$$

where $\mathcal{H}(p||q) = -\int p(\mathbf{y}) \log q(\mathbf{y}) d\mathbf{y}$ is the cross-entropy of p and q and $\mathcal{H}(p) = -\int p(\mathbf{y}) \log p(\mathbf{y}) d\mathbf{y}$ is the entropy of p . $\mathcal{H}(p)$ is constant when comparing different approximations q . The cross-entropy $\mathcal{H}(p||q)$ therefore equals the KL divergence up to a constant, and suffices for comparing approximations q . Because there is no parametric representation of p , corrupted speech samples $\mathbf{y}^{(s)}$ are drawn to approximate it:

$$\mathcal{H}(p||q) = -\int p(\mathbf{y}) \log q(\mathbf{y}) d\mathbf{y} \simeq -\frac{1}{S} \sum_{s=1}^S \log q(\mathbf{y}^{(s)}). \quad (11)$$

Note that when q is the transformed-space sampling method from this paper, for every sample $\mathbf{y}^{(s)}$ another level of sampling occurs inside the evaluation of $q(\mathbf{y}^{(s)})$.

The full-covariance noise and speech Gaussians are both over 24 log-spectral coefficients. The one for the noise is trained directly on the noise audio. The speech distribution is taken from a trained Resource Management system, single-pass re-trained to find Gaussians in the log-spectral domain. (The setup is detailed in section 6.) A low-energy speech component is chosen, to represent the part of the utterance where the low SNR causes recognition errors. The distance between the speech and the noise means, averaged over the log-spectral coefficients, corresponds to a 10 dB SNR. 5000 samples $\mathbf{y}^{(s)}$ are used. For all combinations of speech and noise examined, the relative ordering of the approximation methods is the same.

Figure 2 shows the empirical cross-entropy for different approximations q graphically. DPMC finds the Gaussian that maximises the likelihood, which is the same as minimising cross-entropy. The Gaussian that VTS compensation finds is far from it. IDPMC estimates a mixture of Gaussians from samples with expectation-maximisation. The mixture used here has 8 components trained on 400 000 samples and comes close to the correct 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 is necessary, which quickly becomes impractical.

The transformed-space sampling method introduced in section 4.3 has computational complexity linear in the size of the sample cloud. As the number of samples for the transformed-space sampling method increases, its approximation of $p(\mathbf{y}^{(s)})$ converges to the correct value. This means that the cross-entropy that the line labelled “Sampling” in Figure 2 converges

to is equivalent to a KL divergence of 0. That line, approximately the bottom of the graph, therefore indicates a bound on how well any conceivable model compensation method could match the corrupted speech distribution.

6. Experiments

The usefulness of the cross-entropy to assess compensation methods depends on whether it predicts recogniser performance. This section therefore compares word error rates for compensation methods with their cross-entropy in Figure 2. Since transformed-space sampling needs to be run separately for every observation vector for every speech component, it is too slow to use in a speech recogniser.⁵

The compensation schemes described are evaluated on the 1000 word Resource Management database to which Operations Room noise from the NOISEX-92 database was added at 20 dB and 14 dB. The task contains 109 training speakers reading 3990 sentences, a total of 3.8 hours of data. State-clustered triphone models with 6 components per mixture are built using the HTK RM recipe. All results are averaged over three of the four available test sets, Feb89, Oct89, and Feb91, a total of 30 test speakers and 900 utterances.

The static feature vectors consist of 12 MFCCs, plus the zeroth coefficient, and first- and second-order dynamics. MFCCs are related to log-spectral feature vectors with just a linear transformation (the DCT), so the compensation process is conceptually the same as in the previous sections. Compensation uses extended feature vectors that contain the static feature vectors from a window, and convert them to vectors of statics and dynamics. This yields better performance than using approximations such as the more common continuous time approximation [11]. The extended speech statistics are striped for robustness. The phase factor is assumed independently distributed per time frame. The full-covariance noise model over extended feature vectors is trained directly on the known noise.

Improved modelling of the corrupted speech does not guarantee better discrimination, since speech and noise models are not necessarily the real ones. Table 1 examines recogniser performance, for comparison to Figure 2. 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. This leads to a bigger improvement over the uncompensated system (38.1% to 11.1%) than the modest improvement in cross-entropy would suggest.

However, DPMC, which finds the optimal Gaussian given the speech and noise models, does yield better accuracy (7.4%). IDPMC trains a state-conditional mixture of Gaussians, keeping the number of components constant (“IDPMC”) and increasing it to 12 (“IDPMC+6”). By modelling the distribution better, performance increases to 6.9% and 6.2%. Increasing the number of components further did not improve performance. Since in figure 2 IDPMC comes close to the correct distribution, this gives the best possible performance with the noise model used.

[5] also relates cross-entropy and word error rates for other approximations, such as diagonalising covariances, and assuming $\alpha = 0$. In these cases, the cross-entropy also predicts speech recogniser accuracy. Better modelling of the corrupted speech distribution leads to better performance.

⁵The unoptimised implementation with a sample cloud of 512 would run at roughly 20 million times real time on a modern processor.

Table 1: Word error rates for different compensation schemes.

Compensation	20 dB	14 dB
—	38.1	83.8
VTS	11.1	16.5
DPMC	7.4	13.3
IDPMC	6.9	12.0
IDPMC + 6	6.2	11.1

7. Conclusion

This paper has introduced a new technique for computing the likelihood of a corrupted speech observation vector. It does not use a parametric density, but rather a sampling method. The integral over speech, noise, and phase factor that the likelihood consists of is transformed to allow importance sampling to be applied. As the number of samples goes to infinity, this approximation comes arbitrarily close to the real likelihood. Though the method is too slow to embed in a speech recogniser, it is possible to find the KL divergence from corrupted speech distributions 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 to ideal compensation methods are, and the effect of approximations such as assuming the corrupted speech Gaussian. The KL divergence ranking appears to correspond to the ranking in terms of recognition accuracy.

8. References

- [1] L. Deng, J. Droppo, and A. Acero, “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*, vol. 12, no. 2, pp. 133–143, 2004.
- [2] V. Leutnant and R. Haeb-Umbach, “An analytic derivation of a phase-sensitive observation model for noise robust speech recognition,” in *Proceedings of Interspeech*, 2009, pp. 2395–2398.
- [3] M. J. F. Gales, “Model-based techniques for noise robust speech recognition,” Ph.D. dissertation, Cambridge University, 1995.
- [4] A. Acero, L. Deng, T. Kristjansson, and J. Zhang, “HMM adaptation using vector Taylor series for noisy speech recognition,” in *Proceedings of ICSLP*, vol. 3, 2000, pp. 229–232.
- [5] R. C. van Dalen and M. J. F. Gales, “A theoretical bound for noise-robust speech recognition,” Cambridge University Engineering Department, Tech. Rep. CUED/F-INFENG/TR.648, 2010.
- [6] P. J. Moreno, “Speech recognition in noisy environments,” Ph.D. dissertation, Carnegie Mellon University, 1996.
- [7] B. J. Frey, L. Deng, A. Acero, and T. Kristjansson, “ALGONQUIN: Iterating Laplace’s method to remove multiple types of acoustic distortion for robust speech recognition,” in *Proceedings of Eurospeech*, 2001, pp. 901–904.
- [8] T. T. Kristjansson, “Speech recognition in adverse environments: a probabilistic approach,” Ph.D. dissertation, University of Waterloo, 2002.
- [9] A. Doucet and A. Johansen, “A tutorial on particle filtering and smoothing: fifteen years later,” Department of Statistics, University of British Columbia, Tech. Rep., December 2008. [Online]. Available: http://www.cs.ubc.ca/~arnaud/doucet_johansen_tutorialPF.pdf
- [10] T. A. Myrvoll and S. Nakamura, “Minimum mean square error filtering of noisy cepstral coefficients with applications to ASR,” in *ICASSP*, 2004, pp. 977–980.
- [11] R. C. van Dalen and M. J. F. Gales, “Extended VTS for noise-robust speech recognition,” in *Proceedings of ICASSP*, 2009, pp. 3829–3832.