Transforming Features to Compensate Speech Recogniser Models for Noise

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

Cambridge University Engineering Department, United Kingdom
{rcv25,ff257,mfjg}@eng.cam.ac.uk

Abstract

To make speech recognisers robust to noise, either the features or the models can be compensated. Feature enhancement is often fast; model compensation is often more accurate, because it predicts the corrupted speech distribution. It is therefore able, for example, to take uncertainty about the clean speech into account. This paper re-analyses the recently-proposed predictive linear transformations for noise compensation as minimising the KL divergence between the predicted corrupted speech and the adapted models. New schemes are then introduced which apply observation-dependent transformations in the front-end to adapt the back-end distributions. One applies transforms in the exact same manner as the popular minimum mean square error (MMSE) feature enhancement scheme, and is as fast. The new method performs better on AURORA 2.

Index Terms: speech recognition, noise robustness

1. Introduction

Robustly handling changes in the background noise conditions is a major problem for speech recognition systems. Common approaches are to use either feature enhancement or model compensation techniques. The former traditionally transform the observations to reconstruct a point estimate of the clean speech. The latter transforms the model parameters to accurately represent the distribution of the noise-corrupted speech. Recently, schemes have been proposed that train "predictive" linear transformations from statistics predicted by noise-compensated models [1]. This paper gives a more principled analysis of these predictive linear transforms as minimising the Kullback-Leibler (KL) divergence between the predicted distributions and the models effectively used for decoding.

A number of front-end transformation schemes for noise-robustness that integrate knowledge about observation uncertainty have been proposed. Some use a number of transforms, sending multiple observations to the back-end models at once [2]. Other schemes aim to match model parameters to the expected distribution. They adapt model covariances depending on the observation [3, 4], though this can have issues [5]. Both these approaches are computationally more expensive than just feature transformations, which is what the popular minimum mean square error (MMSE) feature enhancement scheme uses. However, MMSE reconstructs just a point estimate of the clean speech. This paper proposes observation-dependent linear feature transformation methods as fast as MMSE, but of an entirely different nature. Based on predictive CMLLR, they aim to minimise the KL divergence to the corrupted speech distributions.

Adaptive (standard) CMLLR [6] is an instance of a general method of adapting a speech recogniser: applying linear transformations to the component distributions. The transformations are estimated with expectation–maximisation. Given a hypothesis for adaptation data $\mathbf{Y} = \{\mathbf{y}_t\}$, the expectation step computes posteriors $\gamma_{t\mathsf{y}}^{(m)}$ for component m at time t. The maximisation step finds the adaptation parameters $\mathcal{A}$ that maximise the expected likelihood of the complete data. Denoting the adapted distribution for component m with $q\left(\mathbf{y}\middle|\, m, \mathcal{A}\right)$,

$$\hat{\mathcal{A}} = \arg\max_{\mathcal{A}} \sum_t \sum_m \gamma_{t\mathsf{y}}^{(m)} \log q\left(\mathbf{y}_t\middle|\, m, \mathcal{A}\right). \tag{1}$$

For constrained maximum-likelihood linear regression (CMLLR) specifically, the form of q is as follows. $\mathcal{A}$ consists of per-base class transformations $\mathcal{A}^{(n)} = \{\mathbf{A}^{(n)}, \mathbf{b}^{(n)}\}$. Every component m is associated with one base class n. The distribution for component m becomes

$$q\left(\mathbf{y}_t\middle|\, m, \mathcal{A}\right) = \left|\mathbf{A}^{(n)}\right| \mathcal{N}\big(\mathbf{A}^{(n)}\mathbf{y}_t + \mathbf{b}^{(n)}\,;\, \boldsymbol{\mu}_{\mathsf{x}}^{(m)},\, \boldsymbol{\Sigma}_{\mathsf{x}}^{(m)}\big), \tag{2}$$

where $\boldsymbol{\mu}_{\mathsf{x}}^{(m)}$ and $\boldsymbol{\Sigma}_{\mathsf{x}}^{(m)}$ are the mean and covariance matrix for component m of the system trained on clean data. An efficient implementation will transform $\mathbf{y}_t$ differently for every base class. It is therefore easy to mistake CMLLR for a feature transformation. However, which CMLLR transform applies depends solely on the back-end component, and not on the observation. Feature enhancement schemes do not normally apply transformations that depend on the back-end.

This paper will first discuss feature- and model-based forms of noise compensation, and take a fresh look at predictive CMLLR. It will then introduce observation-dependent forms of PCMLLR and assess their performance on AURORA 2.

2. Joint distribution-based compensation

Whereas CMLLR is an adaptive method, other methods compensate for the noise environment specifically. Noise models need fewer parameters, and therefore less adaptation data. The noise compensation methods in this paper all derive from the joint distribution of the clean speech $\mathbf{x}$ and the noise-corrupted speech $\mathbf{y}$, modelled as a Gaussian mixture model (GMM):

$$\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix} \sim \sum_n c^{(n)} \mathcal{N}\left(\begin{bmatrix}\boldsymbol{\mu}_{\mathsf{x}}^{(n)}\\ \boldsymbol{\mu}_{\mathsf{y}}^{(n)}\end{bmatrix}, \begin{bmatrix}\boldsymbol{\Sigma}_{\mathsf{x}}^{(n)} & \boldsymbol{\Sigma}_{\mathsf{xy}}^{(n)}\\ \boldsymbol{\Sigma}_{\mathsf{yx}}^{(n)} & \boldsymbol{\Sigma}_{\mathsf{y}}^{(n)}\end{bmatrix}\right). \tag{3}$$

This front-end distribution is found from a clean speech GMM (from the training data) and a noise model (estimated e.g. per utterance). A variant of first-order vector Taylor series (VTS) compensation [7, 8] yields the distributions in (3) for each clean speech component n. This paper uses standard VTS, which produces diagonal matrices $\boldsymbol{\Sigma}_{\mathsf{xy}}^{(n)}$, $\boldsymbol{\Sigma}_{\mathsf{y}}^{(n)}$ in (3). It is also possible to estimate full covariance matrices with extended VTS [9].

In all the schemes that will be discussed, the most computationally expensive part of the estimation is the VTS compensation, to be precise, computing the derivative of the corrupted speech with relation to the clean speech and noise.

This work was partly funded by Toshiba Research Europe Ltd.

The next sections discuss feature enhancement and model compensation derived from the joint distribution. Both schemes use the standard result that from (3) the distribution of the corrupted speech given the clean speech is

$$\mathbf{y}|\,\mathbf{x}, n \sim \mathcal{N}\Big(\boldsymbol{\mu}_{\mathsf{y}}^{(n)} + \boldsymbol{\Sigma}_{\mathsf{yx}}^{(n)}\boldsymbol{\Sigma}_{\mathsf{x}}^{(n)-1}(\mathbf{x} - \boldsymbol{\mu}_{\mathsf{x}}^{(n)}), \\ \boldsymbol{\Sigma}_{\mathsf{y}}^{(n)} - \boldsymbol{\Sigma}_{\mathsf{yx}}^{(n)}\boldsymbol{\Sigma}_{\mathsf{x}}^{(n)-1}\boldsymbol{\Sigma}_{\mathsf{xy}}^{(n)}\Big). \quad (4)$$

2.1. Feature enhancement

A standard approach to reconstruct a clean speech feature vector $\hat{\mathbf{x}}_t$ from the observation $\mathbf{y}_t$ is to find the minimum mean square error (MMSE) estimate [10]. This uses the mean of (4), with $\mathbf{x}$ and $\mathbf{y}$ swapped around:

$$\begin{aligned}\hat{\mathbf{x}}_t &= \mathcal{E}\{\mathbf{x}|\,\mathbf{y}_t\} = \sum_n P(n|\,\mathbf{y}_t)\,\mathcal{E}\{\mathbf{x}|\,n, \mathbf{y}_t\} \\ &= \sum_n P(n|\,\mathbf{y}_t)\Big(\boldsymbol{\mu}_{\mathsf{x}}^{(n)} + \boldsymbol{\Sigma}_{\mathsf{xy}}^{(n)}\boldsymbol{\Sigma}_{\mathsf{y}}^{(n)-1}(\mathbf{y}_t - \boldsymbol{\mu}_{\mathsf{y}}^{(n)})\Big) \\ &= \sum_n P(n|\,\mathbf{y}_t)\big(\mathbf{A}_{\mathsf{mmse}}^{(n)}\mathbf{y}_t + \mathbf{b}_{\mathsf{mmse}}^{(n)}\big), \end{aligned} \quad (5)$$

where

$$\mathbf{A}_{\mathsf{mmse}}^{(n)} = \boldsymbol{\Sigma}_{\mathsf{xy}}^{(n)}\boldsymbol{\Sigma}_{\mathsf{y}}^{(n)-1}; \quad \mathbf{b}_{\mathsf{mmse}}^{(n)} = \boldsymbol{\mu}_{\mathsf{x}}^{(n)} - \mathbf{A}_{\mathsf{mmse}}^{(n)}\boldsymbol{\mu}_{\mathsf{y}}^{(n)}. \quad (6)$$

This can be implemented as a linear interpolation of transformations $\{\mathbf{A}^{(n)}, \mathbf{b}^{(n)}\}$. The interpolation weights are the component posteriors $P(n|\,\mathbf{y}_t) = c^{(n)}p(\mathbf{y}_t|\,n)/p(\mathbf{y}_t)$, computed for every observation.

Alternatively, every front-end component can be associated with a base class of back-end components. The MMSE transformations are applied per base class, similar to CMLLR, to find a base class-specific reconstruction of the clean speech. This will be called "model-MMSE". Decoding uses (2):

$$q(\mathbf{y}_t|\,m, \mathcal{A}) = \big|\mathbf{A}_{\mathsf{mmse}}^{(n)}\big|\mathcal{N}\big(\mathbf{A}_{\mathsf{mmse}}^{(n)}\mathbf{y}_t + \mathbf{b}_{\mathsf{mmse}}^{(n)}\,;\, \boldsymbol{\mu}_{\mathsf{x}}^{(m)},\, \boldsymbol{\Sigma}_{\mathsf{x}}^{(m)}\big). \quad (7)$$

2.2. Joint uncertainty decoding

Model-based noise-robustness methods are predictive: they predict the distribution of the corrupted speech per component. For example, joint uncertainty decoding (JUD) [11] associates every component m of the speech recogniser HMM with one base class n, whose distribution is approximated by front-end component n. It finds compensation for all components in a base class at once. From the distribution in (4), the conditional corrupted speech distribution for component m is

$$\begin{aligned} p(\mathbf{y}_t|\,m) &= \int p(\mathbf{y}_t|\,\mathbf{x}, n)\,p(\mathbf{x}|\,m)d\mathbf{x} \\ &= \big|\mathbf{A}_{\mathsf{jud}}^{(n)}\big|\mathcal{N}\big(\mathbf{A}_{\mathsf{jud}}^{(n)}\mathbf{y}_t + \mathbf{b}_{\mathsf{jud}}^{(n)}\,;\, \boldsymbol{\mu}_{\mathsf{x}}^{(m)},\, \boldsymbol{\Sigma}_{\mathsf{x}}^{(m)} + \boldsymbol{\Sigma}_{\mathsf{bias}}^{(n)}\big), \end{aligned} \quad (8)$$

with

$$\mathbf{A}_{\mathsf{jud}}^{(n)} = \boldsymbol{\Sigma}_{\mathsf{x}}^{(n)}\boldsymbol{\Sigma}_{\mathsf{yx}}^{(n)-1}; \quad \mathbf{b}_{\mathsf{jud}}^{(n)} = \boldsymbol{\mu}_{\mathsf{x}}^{(n)} - \mathbf{A}_{\mathsf{jud}}^{(n)}\boldsymbol{\mu}_{\mathsf{y}}^{(n)}; \quad (9a)$$

$$\boldsymbol{\Sigma}_{\mathsf{bias}}^{(n)} = \mathbf{A}_{\mathsf{jud}}^{(n)}\boldsymbol{\Sigma}_{\mathsf{y}}^{(n)}\mathbf{A}_{\mathsf{jud}}^{(n)\mathsf{T}} - \boldsymbol{\Sigma}_{\mathsf{x}}^{(n)}. \quad (9b)$$

Since in this paper the covariances of the joint distribution in (3) are diagonal, $\mathbf{A}_{\mathsf{jud}}^{(n)}$ and $\boldsymbol{\Sigma}_{\mathsf{bias}}^{(n)}$ are also diagonal.

Normally, JUD would compensate the parameters for each component according to (8). In this paper, however, it just predicts distributions for training predictive linear transformations.

JUD allows a computationally expensive model compensation method, such as VTS, to transform components of the front-end GMM rather than of the back-end speech recogniser. The number of front-end components gives a trade-off between speed and accuracy. In the limit, if the front-end components are equal to the back-end components, JUD yields exactly the same compensation as the model compensation method would if applied directly on the speech recogniser components.

3. Predictive linear transformations

By taking an adaptive linear transformation method (such as CMLLR) and training it not on data, but on predicted statistics (e.g. from JUD), *predictive* linear transformation methods are obtained. The adaptive optimisation in (1) can be analysed as an approximation to minimising the Kullback-Leibler divergence to the real distributions of the corrupted speech, denoted as $p(\mathbf{y})$, from the distribution of adapted models $q(\mathbf{y}|\,\mathcal{A})$:

$$\begin{aligned}\hat{\mathcal{A}} &= \arg\min_{\mathcal{A}} \mathcal{KL}(p\|\,q) = \arg\min_{\mathcal{A}} \int p(\mathbf{y})\log\frac{p(\mathbf{y})}{q(\mathbf{y}|\,\mathcal{A})}d\mathbf{y} \\ &= \arg\max_{\mathcal{A}} \int p(\mathbf{y})\log q(\mathbf{y}|\,\mathcal{A})d\mathbf{y} = \arg\max_{\mathcal{A}}(-\mathcal{H}(p\|\,q)).\end{aligned} \quad (10)$$

where $\mathcal{H}(p\|\,q)$ is the cross-entropy of p and q.

If the distributions of the corrupted speech and the adapted speech recogniser are assumed mixture models of the form:

$$p(\mathbf{y}) = \sum_m c_{\mathsf{y}}^{(m)}p(\mathbf{y}|\,m); \quad q(\mathbf{y}|\,\mathcal{A}) = \sum_i c_{\mathsf{x}}^{(i)}q(\mathbf{y}|\,i, \mathcal{A}), \quad (11)$$

then the term maximised in (10), the negative cross-entropy, is

$$\begin{aligned} &-\mathcal{H}(p\|\,q) \\ &\quad= \int \sum_m c_{\mathsf{y}}^{(m)}p(\mathbf{y}|\,m)\log\Big(\sum_i c_{\mathsf{x}}^{(i)}q(\mathbf{y}|\,i, \mathcal{A})\Big)d\mathbf{y} \\ &\quad\geq \sum_m \int c_{\mathsf{y}}^{(m)}p(\mathbf{y}|\,m)\log\Big(c_{\mathsf{x}}^{(m)}q(\mathbf{y}|\,m, \mathcal{A})\Big)d\mathbf{y} \\ &\quad= \mathcal{L}(q). \end{aligned} \quad (12)$$

$\mathcal{L}(q)$ is a lower-bound on the negative cross-entropy that assumes that the mixture component indices in both mixture distributions match, i.e. $m = i$. (In practice, this is often a reasonable assumption, as when the corrupted speech distributions are derived from the clean speech distributions.) The lower bound can be maximised as a proxy for minimising the Kullback-Leibler divergence.

The occupancies of the adapted speech recogniser components $c_{\mathsf{x}}^{(m)}$ drop out when $\mathcal{L}(q)$ is optimised with respect to $\mathcal{A}$. The corrupted speech weights $c_{\mathsf{y}}^{(m)}$, on the other hand, can be predicted from the training data occupancy counts $\gamma_{\mathsf{x}}^{(m)}$. When the transformation $\mathcal{A}$ is made up of per-base class transformations $\mathcal{A}^{(n)}$ that affect a subset of components, they can be optimised separately, with $p(\mathbf{y}|\,m)$ predicted by e.g. (8):

$$\begin{aligned}\hat{\mathcal{A}}^{(n)} &= \arg\max_{\mathcal{A}^{(n)}} \sum_{m\in\mathfrak{r}_n} \gamma_{\mathsf{x}}^{(m)} \int p(\mathbf{y}|\,m)\log q\Big(\mathbf{y}\Big|\,m, \mathcal{A}^{(n)}\Big)d\mathbf{y} \\ &= \arg\min_{\mathcal{A}^{(n)}} \sum_{m\in\mathfrak{r}_n} \gamma_{\mathsf{x}}^{(m)}\mathcal{KL}\big(p^{(m)}\big\|q^{(m)}\big). \end{aligned} \quad (13)$$

This makes explicit what predictive linear transformations, introduced in [1], optimise: they aim to minimise the weighted component-for-component KL divergence between the predicted corrupted speech distributions and the adapted models.

3.1. Predictive CMLLR

Predictive CMLLR (PCMLLR) is a version of CMLLR (see section 1) that is trained on predicted corrupted speech distributions. The form of its adaptation is exactly the same as (2). When this expression is substituted into (13), the process of estimating remains the same as for adaptive CMLLR, but its statistics, normally taken from adaptation data, are replaced by pseudo-statistics. These are expected values, which in this paper are derived from JUD. For example, the expectation of y_i for component m can be expressed as

$$\mathcal{E}\left\{ y_i | m \right\} = \left(\mu_{\mathsf{x}i}^{(m)} - b_{\mathsf{jud}i}^{(n)}\right) / a_{\mathsf{jud}i}^{(n)}. \qquad (14)$$

The statistics then become very fast to find online, because much of them can be cached. To save space, this is not derived here, but see [1, 12] for details.

4. Component-independent PCMLLR

The form of decoding with PCMLLR is the same as the one for CMLLR, in (2). Therefore, which transformation $\mathcal{A}^{(n)}$ is chosen from the set of transformations $\mathcal{A}$ depends on the component. However, it is also possible to find a transformation that takes the observation into account when minimising the KL divergence from the effective decoding distribution to the predicted noise-corrupted speech distribution. This could make the transform more appropriate for the acoustic region that the observation is in. The next sections will introduce methods of finding a component-independent transformation $\hat{\mathcal{A}}_t$ at each time instance t. The distribution for component m becomes

$$q\left(\mathbf{y}_t | m\right) = \left|\hat{\mathbf{A}}_t\right| \mathcal{N}\left(\hat{\mathbf{A}}_t \mathbf{y}_t + \hat{\mathbf{b}}_t \,;\, \boldsymbol{\mu}_{\mathsf{x}}^{(m)},\, \boldsymbol{\Sigma}_{\mathsf{x}}^{(m)}\right). \qquad (15)$$

4.1. Estimating a global transform

The simplest scheme for estimating a global transform does not depend on the observation. A global transform is estimated that minimises the KL divergence to the predicted corrupted speech distribution. The optimisation in (13) then becomes

$$\hat{\mathcal{A}} = \arg\min_{\mathcal{A}} \sum_m \gamma_{\mathsf{x}}^{(m)} \mathcal{KL}\left(p^{(m)} \| q^{(m)}\right). \qquad (16)$$

Note that the predicted distribution $p^{(m)}$ is given by $p\left(\mathbf{y} | m\right)$ in (8) with the compensation depending on the base class. This scheme, using the prior predicted distribution $p\left(\mathbf{y}\right)$ to estimate one transform that does not change depending on the observation, will be called "global PCMLLR".

However, given an observation $\mathbf{y}_t$, it is possible to estimate a posterior distribution for $\mathbf{y}$, $p\left(\mathbf{y} | \mathbf{y}_t\right)$, that more closely reflects the distribution that $\mathbf{y}_t$ is likely to have been drawn from. A scheme that estimates one PCMLLR transform from the posterior predicted distribution will be called "observation-trained PCMLLR". To do this robustly, only the weights of the front-end GMM $c^{(n)}$ are replaced by the posteriors $P\left(n | \mathbf{y}_t\right)$. The posterior distribution of $\mathbf{y}$ then becomes

$$p\left(\mathbf{y} | \mathbf{y}_t\right) = \sum_n P\left(n | \mathbf{y}_t\right) p\left(\mathbf{y} | n\right). \qquad (17)$$

Each of the components m of the back-end GMM is weighted or de-weighted by the same amount as its associated front-end component n, so that (16) becomes

$$\hat{\mathcal{A}}_t = \arg\min_{\mathcal{A}} \sum_n \frac{P\left(n | \mathbf{y}_t\right)}{c^{(n)}} \sum_{m \in \mathcal{I}_n} \gamma_{\mathsf{x}}^{(m)} \mathcal{KL}\left(p^{(m)} \| q^{(m)}\right). \qquad (18)$$

$\hat{\mathcal{A}}_t$ in (18) is retrained for every feature vector. This seems computationally expensive at first, but given that back-end components are weighted with a whole base class at once, the necessary statistics are weighted sums of per-base class statistics. The number of base classes is normally only a fraction of the number of back-end components, so this is fast.

4.2. Posterior-weighting transforms

Rather than estimating a global transform, it is possible to estimate a set of transforms appropriate to regions of the acoustic space with PCMLLR, and from those construct a transform observation-specific. A simple way of doing this is to pick the transform associated with the most likely front-end component:

$$\hat{\mathcal{A}} = \mathcal{A}^{(n_t^*)}; \qquad n_t^* = \arg\max_n P\left(n | \mathbf{y}_t\right). \qquad (19)$$

This scheme, "hard-decision PCMLLR", yields a piecewise linear transformation of the feature space.

A more sophisticated approach is similar to front-end CMLLR [11]. This interpolates the transforms, each optimised to minimise the KL divergence to the predicted back-end distributions in an acoustic region. The front-end component posterior is an approximate measure of how likely an observation is to have been generated by a back-end component associated with front-end component n. The transformation becomes

$$\hat{\mathbf{A}}_t = \sum_n P\left(n | \mathbf{y}_t\right) \mathbf{A}^{(n)}; \quad \hat{\mathbf{b}}_t = \sum_n P\left(n | \mathbf{y}_t\right) \mathbf{b}^{(n)}. \quad (20)$$

This will be called "interpolated PCMLLR". The form of the compensation is then the same as for MMSE, so that decoding is equally fast. However, interpolated PCMLLR is properly viewed as transforming the models rather than reconstructing the clean speech. Post-processing of the transformed observation as if it represented the clean speech is therefore not possible.

5. Results

The performance of the proposed schemes was evaluated on the AURORA 2 task [13]. AURORA 2 is a small vocabulary digit string recognition task, based on the TIDIGITS database with noise artificially added. The clean training data comprises 8440 utterances from 55 male and 55 female speakers. The ETSI front-end was used with 39-dimensional feature vectors consisting of 12 MFCCs appended with the zeroth cepstrum, delta and delta-delta coefficients. The acoustic models were whole word models with 16 emitting states and 3 components per state. Results given here are averaged over the four noise conditions in test set A, but the other test sets (B and C) showed exactly the same trends.

64 base classes were used, which as in [7] is about a tenth of the number of back-end components (546). An initial estimate of the additive noise for each utterance was obtained using the first and last 20 frames. This was then re-estimated for every utterance using the ML VTS-based scheme described in [14], applied on the base class level. This noise model was then used to

VTS-compensate the 64-component clean front-end GMM, producing the joint distribution (3). This scheme for estimating the joint distribution is the same as used for MMSE feature enhancement in [7], with the addition of additive noise estimation and explicit compensation of dynamic parameters.

	SNR					
Scheme	20	15	10	5	0	Avg.
VTS	1.6	2.2	4.3	10.5	28.2	9.4
JUD	1.7	2.7	4.7	11.4	30.4	10.2
PCMLLR	1.5	2.6	5.2	13.5	34.7	11.5
Model-MMSE	33.8	47.5	61.6	77.4	91.7	62.4

Table 1: AURORA*: component-dependent compensation.*

Table 1 has word error rates for component-specific model compensation. VTS is the odd one out in that it compensates every back-end component separately, providing accurate but slow compensation. All other schemes use the joint distribution. JUD actually decodes with the back-end models set to their predicted distributions in (8), trading in some accuracy for speed compared to VTS. The other two schemes, PCMLLR and model-MMSE both use component-dependent transformations. PCMLLR minimises the KL divergence to JUD compensation, whereas model-MMSE reconstructs the clean speech. PCMLLR brings the adapted models close to the JUD-predicted statistics. That model-MMSE fails to provide meaningful compensation highlights the difference in the nature of PCMLLR and MMSE transforms, even though their application is exactly the same. MMSE transforms, which aim to reconstruct the clean speech, have no meaning when applied separately to each base class.

	SNR					
PCMLLR Scheme	20	15	10	5	0	Avg.
Global	2.9	6.5	16.8	40.5	71.8	27.7
Observation-trained	1.4	2.6	5.5	14.2	37.1	12.2
Hard-decision	1.5	3.0	6.5	16.6	40.5	13.6
Interpolated	1.4	2.5	5.0	12.6	32.5	10.8
MMSE	1.5	2.6	5.4	14.4	39.7	12.7

Table 2: AURORA*: component-independent transformation:* PCMLLR*-based schemes and* MMSE*.*

Table 2 shows results of using component-independent transformations. MMSE is the standard feature enhancement scheme that reconstructs a clean speech estimate from the noise-corrupted observation. For higher SNRs, its accuracy is similar to model compensation methods JUD and PCMLLR trained from the same joint distribution. For lower SNRs, however, MMSE's point estimate of the clean speech performs less well than JUD and PCMLLR's compensation of distributions.

Global PCMLLR estimates one transform per noise condition, and is the baseline for component-independent PCMLLR. Observation-trained PCMLLR adapts the predicted corrupted speech distribution for every observation and estimates an appropriate transform. This yields accurate compensation, which for lower SNRs shows the advantages of model compensation.

Hard-decision PCMLLR uses the same transforms as PCMLLR, but picks one based on the feature vector rather than on the back-end component. This simple scheme yields a piecewise linear transformation of the feature space. Interpolated PCMLLR performs better, by interpolating the transforms weighted by the front-end posterior. Interpolated PCMLLR outperforms all other derivatives of PCMLLR, PCMLLR itself, and MMSE, which uses the same form of decoding. The ingredients for its performance are twofold. First, the PCMLLR transforms are trained to minimise the KL divergence of the adapted models to the JUD-predicted corrupted speech distributions in an acoustic region. Secondly, the interpolation smoothly moves between transformations that are appropriate for the acoustic region. PCMLLR itself applies component-dependent transformations independently of the feature vector.

6. Conclusion

This paper has presented a new analysis of predictive linear transformations for noise robustness. They minimise the KL divergence from the adapted models to the predicted corrupted speech distributions. This is a markedly different objective than feature enhancement schemes have, which is to find a point estimate of the clean speech vector. A number of approaches have been proposed that apply model compensation, but do so by transforming only the feature vector. Most notably, interpolated PCMLLR offers the same form of decoding as MMSE feature enhancement, but better accuracy on AURORA 2.

7. References

[1] M. J. F. Gales and R. C. van Dalen, "Predictive linear transforms for noise robust speech recognition," in *Proceedings of* ASRU, 2007, pp. 59–64.

[2] V. Stouten, H. Van hamme, and P. Wambacq, "Accounting for the uncertainty of speech estimates in the context of model-based feature enhancement," in *Proceedings of* ICSLP, 2004, pp. 105–108.

[3] J. A. Arrowood and M. A. Clements, "Using observation uncertainty in HMM decoding," in *Proceedings of* ICSLP, 2002, pp. 1561–1564.

[4] J. Droppo, A. Acero, and L. Deng, "Uncertainty decoding with SPLICE for noise robust speech recognition," in *Proceedings of* ICASSP, 2002, pp. 829–832.

[5] H. Liao and M. Gales, "Issues with uncertainty decoding for noise robust speech recognition," *Speech Communication*, vol. 50, no. 4, pp. 265–277, 2008.

[6] M. J. F. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer Speech and Language*, vol. 12, no. 2, pp. 75–98, 1998.

[7] V. Stouten, H. Van hamme, and P. Wambacq, "Joint removal of additive and convolutional noise with model-based feature enhancement," in *Proceedings of* ICASSP, 2004, pp. 949–952.

[8] H. Xu, L. Rigazio, and D. Kryze, "Vector Taylor series based joint uncertainty decoding," in *Proceedings of Interspeech*, 2006, pp. 1125–1128.

[9] R. C. van Dalen and M. J. F. Gales, "Extended VTS for noise-robust speech recognition," in *Proceedings of* ICASSP, 2009, pp. 3829–3832.

[10] Y. Ephraim, "A mimimum mean square error approach for speech enhancement," in *Proceedings of* ICASSP, 1990, pp. 829–832.

[11] H. Liao and M. J. F. Gales, "Uncertainty decoding for noise robust speech recognition," in *Proceedings of Interspeech*, 2005.

[12] F. Flego and M. J. F. Gales, "Incremental predictive and adaptive noise compensation," in *Proceedings of* ICASSP, 2009.

[13] H.-G. Hirsch and D. Pearce, "The AURORA experimental framework for the performance evaluation of speech recognition systems under noise conditions," in *Proceedings of* ASR, 2000, pp. 181–188.

[14] H. Liao and M. J. F. Gales, "Joint uncertainty decoding for robust large vocabulary speech recognition," Cambridge University Engineering Department, Tech. Rep. CUED/F-INFENG/TR.552, November 2006.