Noise Compensation for Subspace Gaussian Mixture Models

Liang Lu[1], *KK Chin*[2], *Arnab Ghoshal*[1], *and Steve Renals*[1]

[1]Centre for Speech Technology Research, University of Edinburgh, Edinburgh, UK
[2]Toshiba Research Europe Ltd, Cambridge Research Laboratory, Cambridge, UK
{liang.lu, a.ghoshal, s.renals}@ed.ac.uk, kkchin@crl.toshiba.co.uk

Abstract

Joint uncertainty decoding (JUD) is an effective model-based noise compensation technique for conventional Gaussian mixture model (GMM) based speech recognition systems. In this paper, we apply JUD to subspace Gaussian mixture model (SGMM) based acoustic models. The total number of Gaussians in the SGMM acoustic model is usually much larger than for conventional GMMs, which limits the application of approaches which explicitly compensate each Gaussian, such as vector Taylor series (VTS). However, by clustering the Gaussian components into a number of regression classes, JUD-based noise compensation can be successfully applied to SGMM systems. We evaluate the JUD/SGMM technique using the Aurora 4 corpus, and the experimental results indicated that it is more accurate than conventional GMM-based systems using either VTS or JUD noise compensation.

1. Introduction

Techniques for speech recognition in noise may perform compensation in the feature domain or in the model domain, although of course there are close relations between the two sets of approaches. In particular, model-based approaches based on vector Taylor series (VTS) have been successfully applied to HMM/GMM systems [1, 2]. However, VTS-based noise compensation is computationally expensive as every Gaussian component in the acoustic model must be adapted, which is a significant problem for systems with a very large number of Gaussians, such as a typical SGMM acoustic model [3]. This problem can be alleviated by joint uncertainty decoding (JUD) [4], in which the whole set of Gaussian components is clustered into a small number of classes using a regression model. The mapping between clean and noise corrupted speech models is shared among the Gaussians belonging to the same regression class.

In this paper, we apply the JUD noise compensation technique to SGMM based acoustic models [3]. In an SGMM, the parameters of each Gaussian component are not estimated directly, but derived from a low dimensional model subspace. This allows a much larger number of Gaussians to be used by each HMM state while limiting the total number of parameters to be estimated. However, this limits the use of VTS-based noise compensation as it operates on the surface GMMs, rather than the compact form, leading to very high computational and memory demands. JUD, on the other hand, compensates the model by estimating feature transformations (except for a covariance bias term). This maintains the compact model structure of SGMMs, and can be more efficient given a smaller regression model. Our experiments on the Aurora 4 corpus indicate that JUD can significantly improve the accuracy of an SGMM system in mismatched conditions introduced by noise, and that this system is more accurate than GMM systems with either VTS or JUD noise compensation.

2. Joint Uncertainty Decoding

In joint uncertainty decoding (JUD), the relationship between clean speech observation $\mathbf{x}$, noisy speech observation $\mathbf{y}$ and model component m can be expressed as:

$$p(\mathbf{y} \mid m) = \int p(\mathbf{x}, \mathbf{y} \mid m) d\mathbf{x} = \int p(\mathbf{y} \mid \mathbf{x}, m) p(\mathbf{x} \mid m) d\mathbf{x}, \tag{1}$$

where $\mathbf{x}$ is viewed as a latent variable, and the conditional probability $p(\mathbf{y} \mid \mathbf{x}, m)$ indicate the effect of noise on clean speech for Gaussian component m. This conditional distribution links the effect of noise with model structure. If the dependency on m is removed, this results in a simplified uncertainty decoding rule, used for many feature domain approaches, for instance SPLICE with uncertainty [5]:

$$p(\mathbf{y} \mid \mathbf{x}, m) \approx p(\mathbf{y} \mid \mathbf{x}). \tag{2}$$

JUD noise compensation performed using (1) is computationally expensive when there are many Gaussian components. To reduce the computational load, the Gaussians may be grouped into a small number of classes based on their acoustic similarities, using the following approximation:

$$p(\mathbf{y} \mid \mathbf{x}, m) \approx p(\mathbf{y} \mid \mathbf{x}, r_m), \tag{3}$$

where r_m denotes the regression class that component m belongs to. We may approximate (1) as:

$$p(\mathbf{y} \mid m) \approx \int p(\mathbf{y} \mid \mathbf{x}, r_m) p(\mathbf{x} \mid m) d\mathbf{x}. \tag{4}$$

The conditional distribution $p(\mathbf{y} \mid \mathbf{x}, r_m)$ is derived from the joint distribution of clean and noise corrupted speech which is assumed to be Gaussian. For the rth regression class

$$p\left(\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \mid r\right) \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_x^{(r)} \\ \boldsymbol{\mu}_y^{(r)} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_x^{(r)} & \boldsymbol{\Sigma}_{yx}^{(r)} \\ \boldsymbol{\Sigma}_{xy}^{(r)} & \boldsymbol{\Sigma}_y^{(r)} \end{bmatrix}\right), \tag{5}$$

which gives the conditional distribution $p(\mathbf{y} \mid \mathbf{x}, r_m)$, with parameters:

$$\boldsymbol{\mu}_{y|x}^{(r)} = \boldsymbol{\mu}_y^{(r)} + \boldsymbol{\Sigma}_{yx}^{(r)} \boldsymbol{\Sigma}_x^{(r)-1} \left(\mathbf{x} - \boldsymbol{\mu}_x^{(r)}\right) \tag{6}$$

$$\boldsymbol{\Sigma}_{y|x}^{(r)} = \boldsymbol{\Sigma}_y^{(r)} - \boldsymbol{\Sigma}_{yx}^{(r)} \boldsymbol{\Sigma}_x^{(r)-1} \boldsymbol{\Sigma}_{xy}^{(r)}. \tag{7}$$

The transformation parameters, $\boldsymbol{\mu}_x^{(r)}$ and $\boldsymbol{\Sigma}_x^{(r)}$, can be estimated from the clean speech model using a regression tree. $\boldsymbol{\mu}_y^{(r)}$, $\boldsymbol{\Sigma}_y^{(r)}$

and the cross covariance $\boldsymbol{\Sigma}_{yx}^{(r)}$ can be obtained by the following mismatch function:

$$\begin{aligned}\mathbf{y}_{\mathbf{s}} &= \mathbf{x}_s + \mathbf{h}_s + \mathbf{C}\log\Big[\mathbf{1} + \exp\left(\mathbf{C}^{-1}(\mathbf{n}_s - \mathbf{x}_s - \mathbf{h}_s)\right) \\ &\quad + 2\boldsymbol{\alpha}\bullet\exp\left(\mathbf{C}^{-1}(\mathbf{n}_s - \mathbf{x}_s - \mathbf{h}_s)/2\right)\Big] \\ &= f(\mathbf{x}_s, \mathbf{n}_s, \mathbf{h}_s, \boldsymbol{\alpha}), \qquad (8)\end{aligned}$$

where the subscript s denotes the static parameters, and $\mathbf{1}$ is the unit vector. Here, $\log(\cdot)$, $\exp(\cdot)$ and $\bullet$ denote the element-wise logarithm, exponentiation and multiplication. $\mathbf{n}_s$ and $\mathbf{h}_s$ are static additive and convolutional noise, respectively. $\mathbf{C}$ is the truncated discrete cosine transform (DCT) matrix, and $\mathbf{C}^{-1}$ indicates its pseudoinverse. $\boldsymbol{\alpha}$ denotes the phase factor [6, 7]. The dynamic parameters can be derived from a continuous time approximation [8]. By marginalising the likelihood in (4), the likelihood of corrupted speech for the m_{th} component can thus be approximated as:

$$p(\mathbf{y} \mid m) \approx |\mathbf{A}^{(r)}|\mathcal{N}\left(\mathbf{A}^{(r)}\mathbf{y} + \mathbf{b}^{(r)}; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m + \boldsymbol{\Sigma}_b^{(r)}\right). \qquad (9)$$

The JUD transformation parameters are obtained as:

$$\mathbf{A}^{(r)} = \boldsymbol{\Sigma}_x^{(r)}\boldsymbol{\Sigma}_{yx}^{(r)-1} \qquad (10)$$

$$\mathbf{b}^{(r)} = \boldsymbol{\mu}_x^{(r)} - \mathbf{A}^{(r)}\boldsymbol{\mu}_y^{(r)} \qquad (11)$$

$$\boldsymbol{\Sigma}_b^{(r)} = \mathbf{A}^{(r)}\boldsymbol{\Sigma}_y^{(r)}\mathbf{A}^{(r)T} - \boldsymbol{\Sigma}_x^{(r)} \qquad (12)$$

The transforms are computed for each regression class, and applied to the Gaussians belonging to the same class in the feature domain[1].

As in standard noise compensation, in equation (8) additive noise is modelled by a single Gaussian $\mathbf{n} \sim \mathcal{N}(\boldsymbol{u}_n, \boldsymbol{\Sigma}_n)$, and the convolutional noise is assumed to be constant $\mathbf{h} = \boldsymbol{\mu}_h$. The mismatch function is highly nonlinear, which makes it difficult to derive the parameters for the noise corrupted speech $\mathbf{y}$. In this work, we use a first order VTS approximation [1] to linearise the mismatch function around the expansion point $\{\boldsymbol{\mu}_{xs}^{(r)}, \boldsymbol{\mu}_{hs}, \boldsymbol{\mu}_{ns}\}$ which results in:

$$\begin{aligned}\mathbf{y}_s \mid r &\approx f(\boldsymbol{\mu}_{xs}^{r}, \boldsymbol{\mu}_{hs}, \boldsymbol{\mu}_{ns}, \boldsymbol{\alpha}) + \mathbf{G}^{(r)}\left(\mathbf{x}_s - \boldsymbol{\mu}_{xs}^{(r)}\right) \\ &\quad + \left(\mathbf{I} - \mathbf{G}^{(r)}\right)(\mathbf{n}_s - \boldsymbol{\mu}_{ns}). \qquad (13)\end{aligned}$$

$\mathbf{G}^{(r)}$ denotes the Jacobian matrix $\frac{\partial f(\cdot)}{\partial \mathbf{x}_s}|_{\boldsymbol{\mu}_{xs}^{(r)}, \boldsymbol{\mu}_{hs}, \boldsymbol{\mu}_{ns}}$. Other approaches include data-driven parallel model combination (DPMC) [9] which draws samples from clean speech and the noise distribution to derive the noisy samples, and higher order VTS [10, 11]. These approaches are normally more expensive, but can result in more accurate speech recognition [12].

3. Joint Uncertainty Decoding for SGMMs

In the SGMM acoustic model [3], the HMM state is modelled as:

$$P(\mathbf{y}_t \mid j) = \sum_{k=1}^{K_j} c_{jk} \sum_{i=1}^{I} w_{jki}\mathcal{N}(\mathbf{y}_t \mid \boldsymbol{\mu}_{jki}, \boldsymbol{\Sigma}_i) \qquad (14)$$

$$\boldsymbol{\mu}_{jki} = \mathbf{M}_i\mathbf{v}_{jk} \qquad (15)$$

$$w_{jki} = \frac{\exp \mathbf{w}_i^T\mathbf{v}_{jk}}{\sum_{i'=1}^{I}\exp \mathbf{w}_{i'}^T\mathbf{v}_{jk}} \qquad (16)$$

where t denotes the time frame, j the HMM state index, k the sub-state index [3], I the number of Gaussians, and K_j the number of sub-states in state j. c_{jk} is a sub-state mixture coefficient and $\boldsymbol{\Sigma}_i$ is the i-th covariance matrix. $\mathbf{v}_{jk} \in \mathbb{R}^S$ is referred to as the sub-state vector, where S denotes the subspace dimension. The matrices $\mathbf{M}_i$ and the vectors $\mathbf{w}_i$ span the model subspaces for Gaussian means and weights respectively, and are used to derive the GMM parameters given sub-state vectors (equations (15) and (16)).

The output densities are initialised using a universal background model (UBM), consisting of I full covariance Gaussian components, which is shared between the states and substates [3]. The UBM defines a partitioning of the acoustic space into I regions, each of which is modeled using a Gaussian with parameters $\mathbf{M}_i$ and $\mathbf{w}_i$, and covariance $\boldsymbol{\Sigma}_i$. The UBM is also used during likelihood evaluation for state j (equations (14)–(16)) to select a reduced set of the most likely Gaussian components for evalaution, rather than evaluating all $K_j \times I$ surface Gaussian components. In other words, the UBM itself provides a clustering of the surface Gaussians based on acoustic similarity which is suitable for JUD noise compensation of SGMMs as discussed below.

3.1. Noise compensation with JUD

JUD noise compensation is performed in the feature domain with only a bias term for covariance. This means the SGMM acoustic model does not need to be expanded (equations (15) and (16)), thus maintaining its compact form. In addition, since JUD does not transform the acoustic model parameters for each Gaussian individually (unlike VTS), the computation is relatively cheap, especially when the number of regression classes is small. To obtain an appropriate regression model for JUD, it would be possible to apply a clustering algorithm to the surface Gaussian components in an SGMM acoustic model, as for a conventional GMM-based system. However, SGMMs have a large number of components (6.4 million in our experiment), so such an approach would be computationally expensive, and would also result in covariance matrices that depend on the regression class rather than being globally shared. This will considerably increase the computation for decoding. Given this, we use the UBM as the regression model which circumvents these issues. Using JUD transforms, the likelihood becomes:

$$\begin{aligned}P(\mathbf{y}_t \mid j, \mathcal{M}_n) &= \sum_{k=1}^{K_j} c_{jk} \sum_{i=1}^{I} w_{jki}\,|\mathbf{A}^{(i)}| \\ &\quad \times \mathcal{N}\left(\mathbf{A}^{(i)}\mathbf{y}_t + \mathbf{b}^{(i)};\, \boldsymbol{\mu}_{jki}, \boldsymbol{\Sigma}_i + \boldsymbol{\Sigma}_b^{(i)}\right) \qquad (17)\end{aligned}$$

where $\mathbf{A}^{(i)}, \mathbf{b}^{(i)}$ and $\boldsymbol{\Sigma}_b^{(i)}$ are derived from the i_{th} Gaussian in the UBM together with the noise model. $\mathcal{M}_n$ denotes the noise model as $\mathcal{M}_n = \{\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n, \boldsymbol{\mu}_h\}$. The updated covariance is still globally shared, but during decoding, we still need to update the normalisation terms for each utterance. Further computation may be saved by using predictive CMLLR [13] to remove the covariance bias term $\boldsymbol{\Sigma}_b^{(i)}$.

3.2. Noise model estimation

For JUD, the noise model estimation is similar to that used in VTS. Two main optimisation approaches have been proposed:

[1] VTS compensation can be reformulated as equation (9), but with no advantage as the transformation must be computed for each Gaussian.

expectation-maximisation (EM) which treats the noise as a latent variable [14]; and a gradient-based approach [7, 15]. A comparison between the two, in terms of accuracy and convergence rate, can be found in [16]. In this paper, we have used a gradient-based approach. The auxiliary function for the noise model update is

$$\mathcal{Q}(\mathcal{M}_n) = \sum_{jkit} \gamma_{jki}(t) \Bigg[\log|\mathbf{A}^{(i)}| + \log \mathcal{N}\left(\mathbf{A}^{(i)}\mathbf{y}_t + \mathbf{b}^{(i)}; \boldsymbol{\mu}_{jki}, \boldsymbol{\Sigma}_i + \boldsymbol{\Sigma}_b^{(i)}\right)\Bigg] \qquad (18)$$

where $\gamma_{jki}(t) = p\left(j, k, i \mid \mathbf{y}_t\right)$ is the Gaussian component posterior.

The additive and convolutional noise means are updated by taking the derivative of $\mathcal{Q}(\cdot)$ with respect to $\boldsymbol{\mu}_n$ and $\boldsymbol{\mu}_h$ to be zero, and a closed form solution can be obtained. However, this is not the case for the additive noise variance $\boldsymbol{\Sigma}_n$, and we use Newton's algorithm to update it. Denoting $\sigma_{n,d}$ as the d_{th} coefficient of $\boldsymbol{\Sigma}_n$,

$$\hat{\sigma}_{n,d} = \sigma_{n,d} - \zeta \left(\frac{\partial^2 \mathcal{Q}(\cdot)}{\partial^2 \sigma_{n,d}}\right)^{-1} \left(\frac{\partial \mathcal{Q}(\cdot)}{\partial \sigma_{n,d}}\right), \qquad (19)$$

where ζ is the learning rate. Note that in practice, the variance may be negative if (19) is applied directly. To enforce positivity, the logarithm of variance is estimated as in [7].

4. Experiments

JUD noise compensation for SGMMs was evaluated on the Aurora 4 corpus, which is derived from the Wall Street Journal (WSJ0) 5k-word closed vocabulary transcription task. The clean training set contains about 15 hours audio, and Aurora 4 provides a noisy version, which allows multi-condition training (MTR). The test set has 300 utterances from 8 speakers. The first test set "test01" (set A) was recorded using a close talking microphone, similar to the clean training data. "test02" to "test07" (set B) were obtained by adding six different types of noise, with randomly selected SNRs ranging from 5dB to 15dB to set A. "test08" (set C) was recording using a desk-mounted secondary microphone and the same type of noise was added to this set which gives "test09" to "test14" (set D). In the following experiments, we used 39 dimensional feature vectors comprising 12th order mel frequency cepstral coefficients (MFCCs), and their first and second derivatives. We used the standard WSJ 5k bigram language model.

4.1. GMM-based systems

Table 1 shows the results of VTS and JUD noise compensation on a conventional GMM system, without the phase term (ie $\boldsymbol{\alpha} = \mathbf{0}$). Here, the clean and MTR models each have about 3.1k triphone states, each speech state modelled by 16 Gaussians while the silence state model uses 32 Gaussians. As expected, the performance of clean model is very poor on noisy testing data, whereas the MTR model can alleviate the mismatch and resulting in significant improvements in accuracy, on average. For the JUD system, we used a regression model with 112 Gaussians, where 48 were used for silence and 64 for speech derived using two separate regression trees. We also carried out VTS-based noise compensation for comparison, which can be viewed as JUD when each Gaussian component corresponds to a regression class.

Table 1: WERs of noise compensation by VTS and JUD on GMM systems with $\boldsymbol{\alpha} = \mathbf{0}$.

Methods	A	B	C	D	Avg
Clean model	7.7	56.6	46.7	72.8	**59.3**
MTR model	12.7	18.6	31.7	36.8	**26.9**
VTS-init	8.7	22.4	43.0	48.0	33.9
+ 1st iter	7.1	15.8	17.3	28.6	20.8
+ 2nd iter	7.3	14.8	12.1	24.8	**18.3**
JUD-init	8.4	23.8	42.6	47.1	34.0
+1st iter	7.2	17.3	24.1	31.8	23.3
+2nd iter	7.0	16.6	16.3	28.7	**21.1**

Table 2: WERs of noise compensation by JUD on SGMM systems with $\boldsymbol{\alpha} = \mathbf{0}$.

Methods	A	B	C	D	Avg
Clean model	5.2	58.2	50.7	72.1	**59.9**
MTR model	6.8	15.2	18.6	32.3	**22.2**
JUD-init	5.5	20.6	36.8	45.6	31.4
+1st iter	5.3	15.3	25.3	32.0	22.5
+2nd iter	5.3	14.7	20.7	28.4	**20.3**

The noise model was initialised by the first and last 20 frames of each testing utterance, corresponding to "VTS-init" and "JUD-init" in table 1. The hypotheses generated by the initial decoding were then used to update the noise model, and another decoding pass was conducted, giving results shown as "1st iter". The procedure was repeated to give the results "2nd iter". Table 1 indicates that updating the noise model leads to considerable gains in accuracy for both VTS and JUD. In addition, VTS-based systems consistently outperform their JUD counterparts as expected. However, the computation cost for JUD is much lower than that for VTS. The lowest word error rate (WER) given by VTS is 18.3% which is comparable to 17.8% reported in [17] with a similar system configuration, and that for JUD is 21.1% which is a little better than 22.2% in [18].

4.2. SGMM-based systems

We used $I = 400$ components in the UBM and a subspace dimension $S = 40$ in the SGMM-based systems. There were about 3,900 tied triphone states, and about 16,000 substates were used in total, resulting in 6.4 million surface Gaussians. Similar to the GMM-based systems, we separated speech and silence in the regression model, using 100 Gaussians for silence and 300 for speech in the UBM. We found that this separation between speech and silence improve the accuracy for the JUD-based systems. Table 2 gives the baseline results using clean and MTR models. The SGMM system has a lower WER than the GMM system on clean test data (A; 5.2% vs. 7.7%); however, the improvement disappears in noisy conditions. This may indicate that SGMMs do not cope with highly mismatched data better than conventional GMMs, and motivates our work to compensate the SGMMs for the mismatch. The MTR model, on the other hand, gives a lower average WER compared with its GMM counterpart (22.2% vs. 26.9%), as the mismatch is less serious.

We then applied JUD noise compensation to a clean SGMM

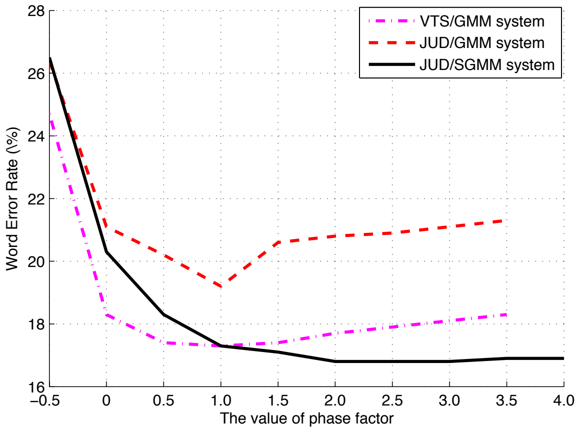

Figure 1: Average WER with respect to the phase term α for both GMM and SGMM system with VTS or JUD style noise compensation. The best result for VTS/GMM is 17.3% ($\boldsymbol{\alpha} = \mathbf{1.0}$), JUD/GMM is 19.2% ($\boldsymbol{\alpha} = \mathbf{1.0}$) and JUD/SGMM is 16.8% ($\boldsymbol{\alpha} = \mathbf{2.5}$).

acoustic model. Table 2 shows the results without the phase term, i.e. $\boldsymbol{\alpha} = \mathbf{0}$. Again, the noise model is initialised by the first and last 20 frames of each utterance, and then updated by the algorithm described in section 3.2. The results show that JUD compensation lead to lower WERs for SGMM systems in mismatched condition, and using a two-pass decoding, we achieve 20.3% WER, which is about 2% absolute lower than the MTR model, but is higher than the VTS/GMM system.

We then investigated a non-zero phase term. As an initial evaluation, we do not estimate the value of $\boldsymbol{\alpha}$ (as in [6]) but set all the coefficients of $\boldsymbol{\alpha}$ empirically to be a fixed value [7]. As a comparison, the phase factor is also tested for GMM-based VTS and JUD system. Figure 1 graphs the average WERs. We find that the phase factor significantly affects both VTS and JUD compensation for GMM and for SGMM systems, consistent with previously reported results [6,7]. The phase factor has a large effect on the JUD/SGMM system: tuning $\boldsymbol{\alpha}$ achieves 16.8% WER, significantly lower than the baseline (20.3%), also lower than the best performance of VTS/GMM by 0.5% absolute. Possible reasons for this improvement may be the correlations between noise and speech captured by the phase factor, and the systematic bias introduced by the VTS linearisation error (equation (13)) [6, 7]. In addition, $\boldsymbol{\alpha} = \mathbf{1}$ corresponds to magnitude domain compensation, which outperforms the power domain compensation ($\boldsymbol{\alpha} = \mathbf{0}$) [19].

5. Conclusion and Future Work

This paper addresses robust speech recognition based on subspace Gaussian mixture models (SGMMs) using joint uncertainty decoding (JUD) noise compensation. We used the UBM as the regression model for JUD clustering, and have investigated noise model estimation based on this configuration. We also discussed the impact of phase factors for noise compensation. Based on the Aurora 4 dataset, we show that JUD can be successfully applied to SGMM-based systems to compensate for acoustic mismatch introduced by noise. In addition, by empirically tuning the value of the phase factors, we observe significant reductions in WER for the JUD/SGMM system, compared with the VTS/GMM and JUD/GMM systems.

Future work may include analytical determination of the phase factor [6, 20], using higher order VTS to improve the approximation accuracy [10, 11], and using extended VTS to obtain a better estimate of the dynamic coefficients [21].

6. Acknowledgements

The work was mostly done during the first author visiting Toshiba Cambridge Research Lab. Thanks to K. Knill, J. Latorre and M. Akamine for arranging the visit. The research was supported by EU FP7 Programme under grant agreement number 213850 (SCALE), and by EPSRC Programme Grant EP/I031022/1 (Natural Speech Technology).

7. References

[1] PJ Moreno, B Raj, and RM Stern, "A vector Taylor series approach for environment-independent speech recognition," in *Proc. ICASSP*. IEEE, 1996, vol. 2, pp. 733–736.

[2] A Acero, L Deng, T Kristjansson, and J Zhang, "HMM adaptation using vector Taylor series for noisy speech recognition," in *Proc. ICSLP*, 2000.

[3] D Povey, L Burget, M Agarwal, P Akyazi, F Kai, A Ghoshal, O Glembek, N Goel, M Karafiát, A Rastrow, RC Rose, P Schwarz, and S Thomas, "The subspace Gaussian mixture model—A structured model for speech recognition," *Computer Speech & Language*, vol. 25, no. 2, pp. 404–439, 2011.

[4] H. Liao and MJF Gales, "Joint uncertainty decoding for noise robust speech recognition," in *Proc. INTERSPEECH*. Citeseer, 2005.

[5] J Droppo, A Acero, and L Deng, "Uncertainty decoding with SPLICE for noise robust speech recognition," in *Proc. ICASSP*. IEEE, 2002.

[6] 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.

[7] J Li, L Deng, D Yu, Y Gong, and A Acero, "A unified framework of HMM adaptation with joint compensation of additive and convolutive distortions," *Computer Speech & Language*, vol. 23, no. 3, pp. 389–405, 2009.

[8] RA Gopinath, MJF Gales, PS Gopalakrishnan, S Balakrishnan-Aiyer, and MA Picheny, "Robust speech recognition in noise—Performance of the IBM continuous speech recogniser on the ARPA noise spoke task," in *Proc. ARPA Workshops Spoken Lang. Syst. Technol.*, 1995, pp. 127–130.

[9] MJF Gales, *Model-based techniques for noise robust speech recognition*, Ph.D. thesis, Cambridge University, 1995.

[10] H Xu and KK Chin, "Joint uncertainty decoding with the second order approximation for noise robust speech recognition," in *Proc. ICASSP*. IEEE, 2009, pp. 3841–3844.

[11] J Du and Q Huo, "A feature compensation approach using high-order vector Taylor series approximation of an explicit distortion model for noisy speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 8, pp. 2285–2293, 2011.

[12] H Xu and KK Chin, "Comparison of estimation techniques in joint uncertainty decoding for noise robust speech recognition," in *Proc. INTERSPEECH*, 2009, pp. 2403–2406.

[13] MJF Gales and RC Van Dalen, "Predictive linear transforms for noise robust speech recognition," in *Proc. ASRU*. IEEE, 2007, pp. 59–64.

[14] DY Kim, C Kwan Un, and NS Kim, "Speech recognition in noisy environments using first-order vector Taylor series," *Speech Communication*, vol. 24, no. 1, pp. 39–49, 1998.

[15] H Liao, *Uncertainty decoding for noise robust speech recognition*, Ph.D. thesis, University of Cambridge, 2007.

[16] Y Zhao and BH Juang, "A comparative study of noise estimation algorithms for VTS-based robust speech recognition," in *Proc. INTERSPEECH*, 2010.

[17] YQ Wang and MJF Gales, "Speaker and noise factorisation on the Aurora 4 task," in *Proc. ICASSP*. IEEE, 2011, pp. 4584–4587.

[18] F Flego and MJF Gales, "Factor analysis based VTS and JUD noise estimation and compensation," in *Proc. ICASSP*. IEEE, 2011, pp. 4792–4795.

[19] MJF Gales and F Flego, "Discriminative classifiers with adaptive kernels for noise robust speech recognition," *Computer Speech & Language*, vol. 24, no. 4, pp. 648–662, 2010.

[20] V Leutnant and R Haeb-Umbach, "An analytic derivation of a phase-sensitive observation model for noise robust speech recognition," in *Proc. INTERSPEECH*, 2009, pp. 2395–2398.

[21] RC van Dalen and MJF Gales, "Extended VTS for noise-robust speech recognition," in *Proc. ICASSP*. IEEE, 2009, pp. 3829–3832.