Noise adaptive training for subspace Gaussian mixture models

Liang Lu, Arnab Ghoshal, and Steve Renals

Centre for Speech Technology Research, University of Edinburgh, Edinburgh, UK

{liang.lu, a.ghoshal, s.renals}@ed.ac.uk

Abstract

Noise adaptive training (NAT) is an effective approach to normalise environmental distortions when training a speech recogniser on noise-corrupted speech. This paper investigates the model-based NAT scheme using joint uncertainty decoding (JUD) for subspace Gaussian mixture models (SGMMs). A typical SGMM acoustic model has much larger number of surface Gaussian components, which makes it computationally infeasible to compensate each Gaussian explicitly. JUD tackles this problem by sharing the compensation parameters among the Gaussians and hence reduces the computational and memory demands. For noise adaptive training, JUD is reformulated into a generative model, which leads to an efficient expectation-maximisation (EM) based algorithm to update the SGMM acoustic model parameters. We evaluated the SGMMs with NAT on the Aurora 4 database, and obtained higher recognition accuracy compared to systems without adaptive training.
Index Terms: adaptive training, noise robustness, joint uncertainty decoding, subspace Gaussian mixture models

1. Introduction

Modern state-of-the-art automatic speech recognition (ASR) systems are normally trained on a large amount of heterogeneous acoustic data recorded from different speakers and in various environmental conditions. This induces nuisance variability and brings in more uncertainty to the acoustic data, and hence reduces the recognition accuracy of an ASR system. Adaptive training is an effective technique to normalise such variability. A typical example is speaker adaptive training (SAT) [1], in which speaker-dependent transformations are trained jointly with the acoustic model parameters in order to account for speaker-related variability. The canonical acoustic model trained in this fashion is a better model for the phonetic variabilities in the acoustic data. Similar adaptive training schemes have also been proposed to normalise the variability induced by environmental noise, which is referred to as noise adaptive training (NAT) [2, 3], including some variants such as irrelevant variability normalisation (IVN) [4] and joint adaptive training (JAT) [5].

The application of NAT depends on the particular choice of the noise compensation algorithms, which may be either feature-domain or model-domain. Several approaches of this nature have been proposed, each with specific strengths and weaknesses. For instance, the vector Taylor series (VTS) [6] and model-based joint uncertainty decoding (JUD) [7] approaches rely on a mismatch function that models the relationship between clean and noise corrupted speech. Using such a mismatch function has the advantage that the required amount of adaptation data is small, which is suitable for rapid adaptation. But its applicability is limited to spectral or cepstral features. SPLICE [2, 8] and front-end JUD [9] remove this constraint by learning a mapping between clean and noisy speech from stereo (both noisy and clean) training data. However, stereo data is normally hard to obtain, and it may not generalise well to unseen noise conditions. Noisy constrained maximum likelihood linear regression (NCMLLR) [10], which is a purely data-driven method, is more flexible from this perspective. It relies neither on a mismatch function (as with VTS or JUD), nor on having stereo training data (as with SPLICE), but estimates the noise compensation transformations using the maximum likelihood (ML) criterion for each homogeneous block of acoustic data. However, it requires a larger amount of training data to achieve good performance, and hence it is not suitable for rapid adaptation.

Previously we extended JUD-based noise compensation to subspace Gaussian mixture models (SGMMs) [11, 12]. Due to its compact representation [13], an SGMM acoustic model usually has a much larger number of surface Gaussians, making it impractical to individually compensate each surface Gaussian. JUD provides a practical way to perform noise compensation for SGMMs [12]. In this paper, we apply NAT to SGMMs using JUD transformations. The adaptive training algorithm is derived from the generative nature of the JUD transformation [10], which leads to an efficient EM-based algorithm to update the acoustic model parameters. We have performed experiments using the NAT algorithm on the Aurora 4 dataset and demonstrate the effectiveness of the proposed approach.

2. Joint uncertainty decoding

In joint uncertainty decoding [9], given a noisy speech observation $\mathbf{y}_t$ at time frame t, the likelihood of the (parameters of) model component m is obtained by marginalising out the latent clean speech variable $\mathbf{x}_t$:

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

$$\approx \int p(\mathbf{y}_t \mid \mathbf{x}_t, r) p(\mathbf{x}_t \mid m) d\mathbf{x}_t \tag{2}$$

where r denotes the regression class that component m belongs to, and equation (2) is obtained by using the approximation $p(\mathbf{y}_t|\mathbf{x}_t, m) \approx p(\mathbf{y}_t|\mathbf{x}_t, r)$. By using smaller number of regression classes, this approximation can significantly reduce the computational cost at the expense of slightly worse recognition accuracy [7].

By assuming that the joint distribution of $\mathbf{x}_t$ and $\mathbf{y}_t$ is Gaussian, the analytical form of the marginal likelihood is written:

$$p(\mathbf{y}_t \mid m) \approx |\mathbf{A}^{(r)}| \mathcal{N}\left(\mathbf{A}^{(r)}\mathbf{y}_t + \mathbf{b}^{(r)}; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m + \boldsymbol{\Sigma}_b^{(r)}\right), \tag{3}$$

where $\boldsymbol{\mu}_m$ and $\boldsymbol{\Sigma}_m$ denote the mean and covariance of Gaussian component m, and $\mathcal{T} = \left[\left(\mathbf{A}^{(r)}, \mathbf{b}^{(r)}, \boldsymbol{\Sigma}_b^{(r)}\right), r = 1, \ldots, R\right]$

are referred to as the JUD transformation parameters, computed for each regression class r, and R is the total number of regression classes. These parameters may be estimated from stereo training data as in SPLICE [9], or they may be estimated directly from the noisy data [14, 7] using a similar mismatch function to the one used in standard VTS-based noise compensation [6, 15]. Following [16, 17], we used the generalised mismatch function which introduces the phase factor to capture the correlations between the noise and clean speech when applying JUD to an SGMM acoustic model [11, 12], which can be expressed as

$$\mathbf{y}_t^{(s)} = \mathbf{x}_t^{(s)} + \mathbf{h}_t + \mathbf{C}\log\Big[\mathbf{1} + \exp\Big(\mathbf{C}^{-1}(\mathbf{n}_t - \mathbf{x}_t^{(s)} - \mathbf{h}_t)\Big) + 2\boldsymbol{\alpha} \bullet \exp\Big(\mathbf{C}^{-1}(\mathbf{n}_t - \mathbf{x}_t^{(s)} - \mathbf{h}_t)/2\Big)\Big], \quad (4)$$

where the superscript $^{(s)}$ corresponds to the static coefficients; $\mathbf{1}$ is a vector, with each element set to 1; $\log(\cdot)$, $\exp(\cdot)$ and $\bullet$ denote the element-wise logarithm, exponentiation and multiplication, respectively; $\mathbf{n}_t$ and $\mathbf{h}_t$ are the static parts of the additive and convolutional noise, respectively; $\mathbf{C}$ is the truncated discrete cosine transform (DCT) matrix, with $\mathbf{C}^{-1}$ as its pseudoinverse; and $\boldsymbol{\alpha}$ denotes the phase factor [16, 17].

2.1. Reformulation as a generative model

JUD may also be represented as a generative model for each regression class r [10]:

$$\mathbf{y}_t = \mathbf{H}^{(r)}\mathbf{x}_t + \mathbf{g}^{(r)} + \mathbf{e}_t^{(r)}, \quad \mathbf{e}_t^{(r)} \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{\Phi}^{(r)}\right) \quad (5)$$

where $\mathbf{H}^{(r)}$ is a linear transform, $\mathbf{g}^{(r)}$ denotes the bias term and $\mathbf{e}_t^{(r)}$ is the residual additive noise which is assumed to be Gaussian distributed. From equation (5), the conditional distribution of $\mathbf{y}_t$ given $\mathbf{x}_t$ for each regression class can be obtained as

$$p(\mathbf{y}_t|\mathbf{x}_t, r) = \mathcal{N}\left(\mathbf{y}_t; \mathbf{H}^{(r)}\mathbf{x}_t + \mathbf{g}^{(r)}, \boldsymbol{\Phi}^{(r)}\right). \quad (6)$$

Given this distribution, the original JUD likelihood function (3) can be obtained by substituting equation (6) into (2) by setting the JUD transformation parameters to be $\mathbf{A}^{(r)} = \mathbf{H}^{(r)-1}, \mathbf{b}^{(r)} = -\mathbf{H}^{(r)-1}\mathbf{g}^{(r)}$ and $\boldsymbol{\Sigma}_b^{(r)} = \mathbf{A}^{(r)}\boldsymbol{\Phi}^{(r)}\mathbf{A}^{(r)T}$.

The generative view of JUD is particularly useful, since it makes it possible to estimate the JUD transforms in a data-driven fashion. It is more flexible as it gets rid of the mismatch function (4). For instance, a successful example can be found in [10] which is also known as noisy-CMLLR. Meanwhile, an EM algorithm can also be derived to update the acoustic model parameter for adaptive training as in [10, 18]. This algorithm will be used in this paper for noise adaptive training of SGMMs which will be further discussed in section 3.

2.2. Compensation of SGMMs

In the SGMM acoustic model [13] the GMM parameters of each HMM state are derived from a low-dimensional model subspace. SGMMs have been shown to improve accuracy compared with conventional GMM-based approaches in conversational telephone speech transcription [13], and in multilingual settings [19, 20]. To perform noise compensation of SGMMs with JUD [11, 12], we use the universal background model (UBM) in the SGMM as the regression model, which leads to a simple implementation and computational efficiency. With JUD transforms, the likelihood function for the HMM state j is

$$p(\mathbf{y}_t \mid j, \mathcal{T}) = \sum_{k=1}^{K_j} c_{jk} \sum_{i=1}^{I} w_{jki} \left|\mathbf{A}^{(r)}\right| \times \mathcal{N}\left(\mathbf{A}^{(r)}\mathbf{y}_t + \mathbf{b}^{(r)}; \boldsymbol{\mu}_{jki}, \boldsymbol{\Sigma}_i + \boldsymbol{\Sigma}_b^{(r)}\right) \quad (7)$$

where c_{jk} and w_{jki} are the sub-state and Gaussian component weights, I denotes the number of Gaussians in the UBM, $\boldsymbol{\Sigma}_i$ is the global covariance matrix for the i-th Gaussian, and K_j is the number of sub-states for state j [13]. The Gaussian means and weights are derived as:

$$\boldsymbol{\mu}_{jki} = \mathbf{M}_i\mathbf{v}_{jk}, \quad w_{jki} = \frac{\exp \mathbf{w}_i^T\mathbf{v}_{jk}}{\sum_{i'=1}^{I}\exp \mathbf{w}_{i'}^T\mathbf{v}_{jk}}. \quad (8)$$

Here, $\mathbf{M}_i$ and $\mathbf{w}_i$ are the mean and weight projections, and $\mathbf{v}_{jk}$ is the state vector which is normally low dimensional. The regression class index r will be replaced by the UBM component index i if using the UBM as the regression model for JUD [12].

3. Noise adaptive training

Usually noise compensation is employed on a per-utterance basis [3, 17], since the noise condition is assumed to be fixed for the duration of an utterance. This means that the JUD transformation $\mathcal{T}$ depend on the utterance. However, we omit the utterance index on $\mathcal{T}$ in order to simplify the notation. In what follows, we use $\mathcal{M}$ to denote the SGMM acoustic model parameters.

Noise adaptive training (NAT) of the acoustic model involves joint optimisation of the acoustic model parameters $\mathcal{M}$ and the transformation parameters $\mathcal{T}$. For an SGMM acoustic model, the auxiliary function for NAT is

$$\mathcal{Q}\left(\mathcal{M}, \mathcal{T}; \tilde{\mathcal{M}}, \tilde{\mathcal{T}}\right) = \sum_{jkit} \gamma_{jki}(t) \log\left|\mathbf{A}^{(r)}\right| \times \mathcal{N}\left(\mathbf{A}^{(r)}\mathbf{y}_t + \mathbf{b}^{(r)}; \boldsymbol{\mu}_{jki}, \boldsymbol{\Sigma}_i + \boldsymbol{\Sigma}_b^{(r)}\right) \quad (9)$$

where $\tilde{\mathcal{M}}$ and $\tilde{\mathcal{T}}$ denote the current estimate of the model and transformation parameters, and $\gamma_{jki}(t)$ is the posterior probability, computed using $\tilde{\mathcal{M}}$ and $\tilde{\mathcal{T}}$. This auxiliary function is for a particular training utterance that the transformation parameters $\mathcal{T}$ depend on. The overall auxiliary function for the entire training set is obtained by summing (9) over all utterances.

Directly optimising either $\mathcal{M}$ or $\mathcal{T}$ is computationally demanding, especially for an SGMM, since the auxiliary function is complex. Analogous to SAT [1], a common practice is to interleave the update of $\mathcal{M}$ and $\mathcal{T}$ one after another [3, 5]. In this paper, we adopt the same principle for adaptive training of SGMMs. We have previously detailed the estimation of $\mathcal{T}$ given $\mathcal{M}$ [12]; in this paper, we focus on the estimation of the acoustic model parameters $\mathcal{M}$ given the estimate of the transformation parameters $\mathcal{T}$.

3.1. Optimisation

Two optimisation approaches for the update of the acoustic model parameters $\mathcal{M}$ in NAT have been investigated: second-order gradient-based [5, 3] and EM-based [10].

In the second-order gradient-based approach a particular set of parameters θ in $\mathcal{M}$ is updated by

$$\theta = \tilde{\theta} - \zeta\left[\left(\frac{\partial^2\mathcal{Q}(\cdot)}{\partial^2\theta}\right)^{-1}\left(\frac{\partial\mathcal{Q}(\cdot)}{\partial\theta}\right)\right]_{\theta=\tilde{\theta}} \quad (10)$$

where $\tilde{\theta}$ is the current value of θ, ζ is the learning rate and $\mathcal{Q}(\cdot)$ denotes the auxiliary function (9). Such gradient-based optimisation was used for JUD-GMM systems [5] and for VTS-GMM systems [3]. Depending on the form of Hessian, it may yield faster convergence. However, the drawbacks of this approach are that the computation of the gradient and Hessian terms in (10) can be complex, especially for the SGMM-based acoustic models due to the compact model representation. Furthermore, it is not simple to perform gradient-based optimisation when using a discriminative criterion [18].

The second type of optimisation is based on the EM algorithm, which is derived from viewing the JUD transformation as a generative model (5). This method requires computing sufficient statistics of the expected "pseudo-clean" speech feature $\mathbf{x}_t$, which is obtained by computing its conditional distribution given component m:

$$p(\mathbf{x}_t|\mathbf{y}_t, r, m) = \frac{p(\mathbf{y}_t|\mathbf{x}_t, r)p(\mathbf{x}_t|m)}{\int p(\mathbf{y}_t|\mathbf{x}_t, r)p(\mathbf{x}_t|m)d\mathbf{x}_t}. \quad (11)$$

As shown in [10], an analytical solution can be obtained from (6), which gives the conditional expectations as

$$\mathbb{E}[\mathbf{x}_t|\mathbf{y}_t, r, m] = \tilde{\mathbf{x}}_t^{(rm)} \quad (12)$$

$$\mathbb{E}[\mathbf{x}_t\mathbf{x}_t^T|\mathbf{y}, r, m] = \tilde{\mathbf{\Sigma}}_x^{(rm)} + \tilde{\mathbf{x}}_t^{(rm)}\tilde{\mathbf{x}}_t^{(rm)T} \quad (13)$$

where

$$\begin{aligned}
\tilde{\mathbf{x}}_t^{(rm)} &= \tilde{\mathbf{A}}^{(rm)}\mathbf{y}_t + \tilde{\mathbf{b}}^{(rm)} \\
\tilde{\mathbf{\Sigma}}_x^{(rm)} &= \left(\mathbf{\Sigma}_x^{(m)-1} + \mathbf{\Sigma}_b^{(r)-1}\right)^{-1} \\
\tilde{\mathbf{A}}^{(rm)} &= \tilde{\mathbf{\Sigma}}_x^{(rm)}\mathbf{\Sigma}_b^{(r)-1}\mathbf{A}^{(r)} \\
\tilde{\mathbf{b}}^{(rm)} &= \tilde{\mathbf{\Sigma}}_x^{(rm)}\left(\mathbf{\Sigma}_x^{(m)-1}\boldsymbol{\mu}_x^{(m)} + \mathbf{\Sigma}_b^{(r)-1}\mathbf{b}^{(r)}\right)
\end{aligned}$$

where $\boldsymbol{\mu}_x^{(m)}$ and $\mathbf{\Sigma}_x^{(m)}$ are the mean and covariance of Gaussian component m. Given the expectations, the statistics can be accumulated in the standard fashion to re-estimate the acoustic model parameters. This method makes the implementation much simpler and hence has been used in this work.

3.2. Model update

Using the EM-based NAT, described above, only minor changes are required to be made from the original model estimation formula of the SGMMs presented in [13]. Taking the estimation of the Gaussian mean projection $\mathbf{M}_i$ for instance, the auxiliary function is

$$\mathcal{Q}(\mathbf{M}_i) = tr\left(\mathbf{M}_i^T\mathbf{\Sigma}_i^{-1}\mathbf{Y}_i\right) - \frac{1}{2}tr\left(\mathbf{M}_i^T\mathbf{\Sigma}_i^{-1}\mathbf{M}_i\mathbf{Q}_i\right) \quad (14)$$

where the sufficient statistics $\mathbf{Y}_i$ and $\mathbf{Q}_i$ are now obtained as

$$\mathbf{Y}_i = \sum_{jkt}\gamma_{jki}(t)\mathbb{E}[\mathbf{x}_t|\mathbf{y}_t, r, m]\mathbf{v}_{jk}^T \quad (15)$$

$$\mathbf{Q}_i = \sum_{jkt}\gamma_{jki}(t)\mathbf{v}_{jk}\mathbf{v}_{jk}^T. \quad (16)$$

Note that in an SGMM, the Gaussian component index m is replaced by jki as in (7), and the regression class index r is replaced by i. It also worth emphasising that the posterior probability $\gamma_{jki}(t)$ needs to be computed using the noisy feature vector $\mathbf{y}_t$ using the likelihood function (7) during the adaptive training phase.

Likewise, other types of SGMM acoustic model parameters such as $\mathbf{v}_{jk}$ and $\mathbf{\Sigma}_i$ can be estimated in the same fashion using the expectations of the "pseudo-clean" feature vectors. The EM-based algorithm for NAT is similar to some feature enhancement methods which also estimate $\mathbf{x}_t$ given $\mathbf{y}_t$, e.g. [6]. However, a fundamental difference is that the conditional expectations directly relate to the acoustic model structure as in (12) and (13), while for feature enhancement they are normally derived using a front-end GMM. Due to the closer match to the acoustic model, NAT was found to outperform its feature enhancement counterpart in [21].

Finally, note that the UBM associated with the SGMM acoustic model also needs to be updated during adaptive training. After NAT, the SGMM models the "pseudo-clean" features $\mathbf{x}_t$, while the UBM is originally trained on the noise-corrupted features $\mathbf{y}_t$. Since the UBM provides the regression class for the Gaussian components when applying JUD [12], it should model the same acoustic features as the SGMM. In this work, the UBM is updated using the weighted average of the corresponding Gaussian component in the SGMM as

$$\mathbf{\Sigma}_i^{ubm} = \mathbf{\Sigma}_i \quad (17)$$

$$w_i^{ubm} = \frac{\sum_{jkt}\gamma_{jki}(t)}{\sum_{jkit}\gamma_{jki}(t)} \quad (18)$$

$$\boldsymbol{\mu}_i^{ubm} = \sum_{jkt}\frac{\gamma_{jki}(t)}{\sum_{jkt}\gamma_{jki}(t)}\mathbf{M}_i\mathbf{v}_{jk} \quad (19)$$

where w_i^{ubm}, $\boldsymbol{\mu}_i^{ubm}$ and $\mathbf{\Sigma}_i^{ubm}$ are the weight, mean and covariance matrix for component i in the UBM respectively. Updating the UBM was found to improve the recognition accuracy of the NAT system.

3.3. Training recipe

To sum up, the NAT recipe for an SGMM acoustic model used in this paper is as follows.

1. Initialise the acoustic model $\mathcal{M}$ by the standard maximum likelihood training.
2. For each training utterance, initialise the noise model parameters for $\mathbf{n}_t$ and $\mathbf{h}_t$ in (4).
3. Re-estimate the noise model parameters given $\mathcal{M}$.
4. Obtain the JUD transform $\mathcal{T}$ for each utterance.
5. Given $\mathcal{M}$ and $\mathcal{T}$, compute the posterior probability $\gamma_{jki}(t)$ using (7).
6. Accumulate the statistics using the conditional expectations (12) (13) and update $\mathcal{M}$.
7. Go to step 5 until convergence.
8. Update the UBM using equations (17) - (19).
9. Go to step 2 until the number of iterations is reached.

While this paper focuses on the NAT algorithm for the SGMMs, more details about noise model and JUD transform estimation used in step 2 to step 4 can be found in [12].

4. Experiments

The experiments were performed using the Aurora 4 corpus, which is derived from the Wall Street Journal (WSJ0) 5,000-word (5k) closed vocabulary transcription task. The clean training set is the (WSJ0 SI-84) contains about 15 hours of speech. The test set has 300 utterances from 8 speakers. The first

Table 1: Word error rates (WERs) of SGMM systems with and without noise adaptive training.

Methods	A	B	C	D	Avg
Clean model	5.2	58.2	50.7	72.1	59.9
+JUD	5.1	13.1	12.0	23.2	16.8
MST model	6.8	15.2	18.6	32.3	22.2
+JUD	7.4	13.3	14.7	24.1	17.6
NAT model	6.5	20.3	19.8	39.7	27.6
+JUD	6.1	11.3	11.9	22.4	15.7

test set, set A (`test01`) is clean speech, recorded using a close talking microphone, similar to the training data. Set B (`test02` to `test07`) is obtained by adding six different types of noise, with randomly selected signal-to-noise ratios ranging from 5dB to 15dB, to set A. Set C (`test08`) is recorded using a desk-mounted secondary microphone and hence contains distortions due to reverberation. The same type of noise used for set B is added to this test set to form set D (`test09` to `test14`). Additionally, Aurora 4 also provides a 15 hour noisy version of the training set, which contains speech contaminated by the different noise conditions. This training set is used for multi-style training (MST) as well as noise adaptive training (NAT) of the acoustic models.

In the following experiments, we use 39 dimensional feature vectors derived from 12th order mel frequency cepstral coefficients, plus the zeroth order coefficient (C0), with delta and acceleration features. We use the standard WSJ0 5k bigram language model [22] and the CMU pronunciation dictionary. The SGMM systems have about 3900 tied triphone states, 16,000 sub-states, and $I = 400$ Gaussians in the UBM, which results in 6.4 million surface Gaussians. As mentioned before, the phase-sensitive mismatch function (4) is used to estimate the JUD transforms. Based on our previous findings [12], all the entries in $\boldsymbol{\alpha}$ are empirically set to 2.5 in both training and decoding stages unless otherwise specified.

4.1. Results

The experimental results are given in Table 1 using the clean, MST and NAT acoustic models. The NAT system is trained following the recipe in section 3.3, where we perform 4 iterations in step 7 which yields convergence, and only 1 iteration in step 9. As expected, the MST system is significantly more accurate than the clean trained system without JUD compensation since the mismatch between the training and testing data is reduced. However, with JUD compensation we observe that the clean model is more accurate than MST (16.8% vs. 17.6%). This may be due to the larger variability in the MST model making it less suitable for rapid adaptation towards a particular noise condition using limited adaptation data. The NAT system, on the other hand, normalises the irrelevant variability in the training data using noise dependent JUD transforms. Without JUD in the decoding stage, this model results in higher WER than MST, since it does not match the testing data well. With JUD adaptation, however, it is more accurate than the MST and clean systems with a WER of 15.7%, which is slightly better than the adaptively-trained GMM system using VTS on the same dataset (16.0%) [23].

Previous work on empirically tuning the phase factor $\boldsymbol{\alpha}$ in (4) has shown that it is able to bring significant gains in both VTS- and JUD-based noise robust speech recognition systems

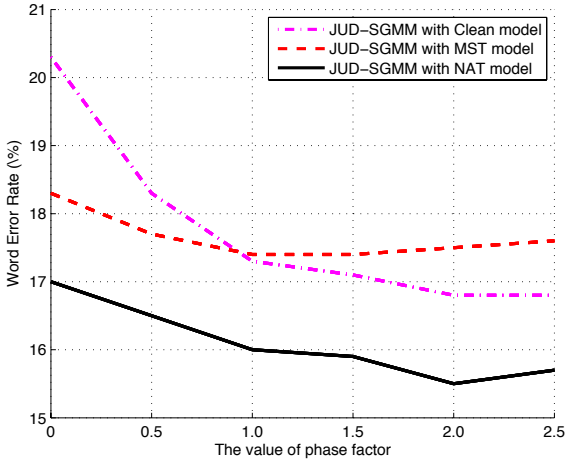

Figure 1: Results of tuning the value of phase factor α_i.

[16, 17, 12]. Interpreted as a phase factor, the values of the elements of $\boldsymbol{\alpha}$ should be in the range $[-1, 1]$ [16]. However, experimental studies have demonstrated that treating $\boldsymbol{\alpha}$ as additional model parameters tuned to mitigate the mismatch between the training and testing data results in improved accuracy [17, 12]. While previous studies on this issue were mainly based on systems trained on clean data [17, 12], we see similar trends with our MST and NAT systems. Figure 1 shows the WER of the systems using the three models by empirically tuning the values of $\boldsymbol{\alpha}$ in the decoding stage as in [17, 12]. It shows that tuning the value of $\boldsymbol{\alpha}$ results in gains for all the three systems, e.g. 15.5% ($\alpha_i = 2.0$) vs. 17.0% ($\alpha_i = 0$) for the NAT system. However, compared to the MST and NAT systems that are trained on multi-condition data, the improvement is much larger for the highly mismatched system that is trained on clean data, e.g. 16.8% ($\alpha_i = 2.0$) vs. 20.3% ($\alpha_i = 0$). These results support the previous argument that tuning $\boldsymbol{\alpha}$ may help to reduce the mismatch between the training and testing conditions. Note that, the results were obtained by tuning $\boldsymbol{\alpha}$ in the decoding phase only; future work will investigate the effect of $\boldsymbol{\alpha}$ on the training stage for NAT system.

5. Conclusions

We have investigated the noise adaptive training (NAT) algorithm for an SGMM acoustic model using multi-condition training data. Our method is based on the joint uncertainty decoding (JUD) noise compensation technique. For adaptive training, an EM-based optimisation algorithm is employed which is derived from reformulating JUD adaptation into a generative model. This algorithm has proven to be simple for implementation, and effective in terms of recognition accuracy. Evaluation was carried out using the Aurora 4 dataset; using NAT, the SGMM system achieved the lowest WER (15.5%) which is considerably better than systems without adaptive training. These experiments are also helpful to understand the effect of phase factor parameter in the mismatch function. Future work will be on applying a discriminative criterion to the adaptively trained system that has been found effective with GMM based systems [18, 24].

Acknowledgement Thanks to the reviewers for their insightful comments. This work is funded by the EPSRC Programme Grant EP/I031022/1 (Natural Speech Technology).

6. References

[1] Y. Anastasakos, J. McDonough, R. Schwartz, and J. Makhoul, "A compact model for speaker-adaptive training," in *Proc. ICSLP*, 1996.

[2] L. Deng, A. Acero, M. Plumpe, and X. Huang, "Large-vocabulary speech recognition under adverse acoustic environments," in *Proc. ICSLP*, 2000.

[3] O. Kalinli, M. Seltzer, J. Droppo, and A. Acero, "Noise adaptive training for robust automatic speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 8, pp. 1889–1091, 2010.

[4] Y. Hu and Q. Huo, "Irrelevant variability normalization based HMM training using VTS approximation of an explicit model of environmental distortions," in *Proc. Interspeech*, 2007, pp. 1042–1045.

[5] H. Liao and M. Gales, "Adaptive training with joint uncertainty decoding for robust recognition of noisy data," in *Proc. ICASSP*. IEEE, 2007.

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

[7] H. Liao, "Uncertainty decoding for noise robust speech recognition," Ph.D. dissertation, University of Cambridge, 2007.

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

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

[10] D. Kim and M. Gales, "Noisy constrained maximum-likelihood linear regression for noise-robust speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 2, pp. 315–325, 2011.

[11] L. Lu, K. Chin, A. Ghoshal, and S. Renals, "Noise compensation for subspace Gaussian mixture models," in *Proc. INTERSPEECH*, 2012.

[12] ——, "Joint uncertainty decoding for noise robust subspace Gaussian mixture models," *IEEE Transactions on Audio, Speech, and Language Processing*, 2013.

[13] D. Povey, L. Burget, M. Agarwal, P. Akyazi, F. Kai, A. Ghoshal, O. Glembek, N. Goel, M. Karafiát, A. Rastrow, R. 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.

[14] H. Xu, L. Rigazio, and D. Kryze, "Vector Taylor series based joint uncertainty decoding," in *Proc. Interspeech*, 2006.

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

[16] 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.

[17] 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.

[18] F. Flego and M. Gales, "Discriminative adaptive training with VTS and JUD," in *Proc. ASRU*. IEEE, 2009, pp. 170–175.

[19] L. Lu, A. Ghoshal, and S. Renals, "Regularized subspace Gaussian mixture models for cross-lingual speech recognition," in *Proc. IEEE ASRU*, 2011.

[20] L. Burget, P. Schwarz, M. Agarwal, P. Akyazi, K. Feng, A. Ghoshal, O. Glembek, N. Goel, M. Karafiát, D. Povey, A. Rastrow, R. Rose, and S. Thomas, "Multilingual acoustic modeling for speech recognition based on subspace Gaussian mixture models," in *Proc. IEEE ICASSP*, 2010, pp. 4334–4337.

[21] J. Li, M. Seltzer, and Y. Gong, "Improvements to VTS feature enhanecement," in *Proc. ICASSP*. IEEE, 2012, pp. 4677–4680.

[22] D. Paul and J. Baker, "The design for the Wall Street Journal-based CSR corpus," in *Second International Conference on Spoken Language Processing*, 1992.

[23] F. Flego and M. Gales, "Factor analysis based VTS discriminative adaptive training," in *Proc. ICASSP*. IEEE, 2012, pp. 4669–4672.

[24] M. 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.