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
{l i a n g . l u , a . g h o s h a l , s . r e n a l s } @ e d . a c . u k

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 re-formulated 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 in the acoustic data which is irrelevant to the task of speech recognition, 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 do noise compensation for SGMMs [12]. In this paper, we study the application of 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 \( y_t \) at time frame \( t \), the likelihood of the (parameters of) model component \( m \) is obtained by marginalising out the latent clean speech variable \( x_t \):

\[
 p(y_t | m) = \int p(x_t, y_t | m) dx_t
 = \int p(y_t | x_t, r) p(x_t | m) dx_t
\]

where \( r \) denotes the regression class that component \( m \) belongs to, and equation (2) is obtained by using the approximation

\[
 p(y_t | x_t, r) \approx p(y_t | x_t, r).
\]

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 \( x_t \) and \( y_t \) is Gaussian, the analytical form of the marginal likelihood is written:

\[
 p(y_t | m) \approx |A^{(r)}||N(\mu_{m}^{(r)} + b^{(r)}; \Sigma_{m} + \Sigma_{e}^{(r)})|.
\]

Here, \( T = \{A^{(r)}, b^{(r)}, \Sigma_{e}^{(r)}\}, r = 1, \ldots, R \) are referred to as the JUD transformation parameters, computed for each re-
gression 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 extended 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

\[
y_t^{(s)} = x_t^{(s)} + h_t + C \log \left[ 1 + \exp \left( C^{-1} (n_t - x_t^{(s)} - h_t) \right) + 2a \cdot \exp \left( C^{-1} (n_t - x_t^{(s)} - h_t/2) \right) \right],
\]

where the superscript \(^{(s)}\) corresponds to the static coefficients; \( I \) is the unit vector; \( \log(\cdot) \), \( \exp(\cdot) \), and \( \cdot \) denote the element-wise logarithm, exponentiation and multiplication, respectively; \( n_t \) and \( h_t \) are the static parts of the additive and convolutional noise, respectively; \( C \) is the truncated discrete cosine transform (DCT) matrix, with \( C^{-1} \) as its pseudoinverse; and \( a \) 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]:

\[
y_t = H^{(r)} x_t + g^{(r)} + n_t^{(r)}, \quad n_t^{(r)} \sim \mathcal{N}(0, \Phi^{(r)})
\]

where \( H^{(r)} \) is a linear transform, \( g^{(r)} \) denote the bias term and \( n_t^{(r)} \) is a Gaussian additive noise. From equation (5), the conditional distribution of \( y_t \) given \( x_t \) for each regression class can be obtained as

\[
p(y_t | x_t, r) = \mathcal{N} \left( y_t; H^{(r)} x_t + g^{(r)}, \Phi^{(r)} \right).
\]

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 \( A^{(r)} = H^{(r)^{-1}}, \Phi^{(r)} = -H^{(r)^{-1}} g^{(r)} \) and \( \Sigma^{(r)} = A^{(r)^{-1}} \Phi^{(r)} A^{(r)^{-T}} \).

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. The likelihood function for the HMM state \( j \) is

\[
p(y_t | j, T) = \sum_{k=1}^{K_j} \sum_{i=1}^{I} w_{jki} A^{(r)} \mathcal{N} \left( y_t; \mathbf{b}^{(r)}; \mu_{jki}, \Sigma_{jki}^{(r)} \right)
\]

where \( c_{jki} \) and \( w_{jki} \) are the sub-state and Gaussian component weights, \( I \) denotes the number of Gaussians in the UBM, \( \Sigma_{jki} \) 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:

\[
\mu_{jki} = M_j v_{jki}, \quad w_{jki} = \frac{\exp w_{jki}^{T} v_{jki}}{\sum_{i'\in I} \exp w_{jki'}^{T} v_{jki'}}.
\]

Here, \( M_j \) and \( w_{jki} \) are the mean and weight projections, and \( v_{jki} \) 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]. 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 \( T \) depend on the utterance. However, we omit the utterance index on \( T \) in order to simplify the notation. In the following sections, we use \( M \) to denote the SGMM acoustic model parameters.

3. Noise adaptive training

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

\[
Q \left( \tilde{M}, \tilde{T}; M, T \right) = \sum_{jkt} \gamma_{jkt}(t) \log |A^{(r)}| \times \mathcal{N} \left( A^{(r)} y_t + b^{(r)}; \mu_{jkt}, \Sigma_{jkt}^{(r)} \right)
\]

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

Directly optimising either \( M \) or \( T \) 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 \( M \) and \( T \) 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 \( T \) given \( M \) [12]; in this paper, we focus on the estimation of the acoustic model parameters \( M \) given the estimate of the transformation parameters \( T \).

3.1. Optimisation

Two optimisation approaches for the update of the acoustic model parameters \( M \) 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 \( \theta \) in \( M \) is updated by

\[
\dot{\theta} = \theta - \zeta \left( \frac{\partial^2 Q(\cdot)}{\partial \theta^2} \right)^{-1} \left( \frac{\partial Q(\cdot)}{\partial \theta} \right)
\]
where \( \hat{\theta} \) is the new value of \( \theta \), \( \zeta \) is the learning rate and \( Q(\cdot) \) 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 do gradient-based optimisation when using a discriminative criteria [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 \( x_t \), which is obtained by computing its conditional distribution given component \( m \):

\[
p(x_t|y_t, r, m) = \frac{p(y_t|x_t, r)p(x_t|m)}{\int p(y_t|x_t, r)p(x_t|m)dx_t}
\]

(11)

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

\[
E[x_t|y_t, r, m] = \hat{x}_t^{(rm)}
\]

(12)

\[
E[x_t x_t^T|y_t, r, m] = \Sigma_x^{(rm)} + \hat{\Sigma}_x^{(rm)}\hat{x}_t^{(rm)}\hat{x}_t^{(rm)T}
\]

(13)

where

\[
\hat{x}_t^{(rm)} = \hat{A}^{(rm)}y_t + \hat{b}^{(rm)}
\]

\[
\hat{A}^{(rm)} = (\Sigma_x^{(m)} - \Sigma_y^{(r)}b^{(r)})^{-1}A^{(r)}
\]

\[
\hat{b}^{(rm)} = \Sigma_x^{(rm)}(\Sigma_x^{(m)} - \Sigma_y^{(r)}b^{(r)})^{-1}(\mu_x^{(m)} + \Sigma_y^{(r)}b^{(r)})
\]

where \( \mu_x^{(m)} \) and \( \Sigma_x^{(m)} \) 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 involves minor changes in the original model estimation formula of the SGMMs presented in [13]. Taking the estimation of the Gaussian mean projection \( \hat{M}_i \), for instance, the auxiliary function is

\[
Q(M_i) = tr \left( M_i^T \Sigma_i^{-1} \bar{Y}_t \right) - \frac{1}{2} tr \left( M_i^T \Sigma_i^{-1} M_i Q_i \right)
\]

(14)

where the sufficient statistics \( \bar{Y}_t \) and \( Q_i \) are now obtained as

\[
\bar{Y}_t = \sum_{jkt} \gamma_{jkt}(t) E[x_t|y_t, r, m] v_{jk}^T
\]

(15)

\[
Q_i = \sum_{jkt} \gamma_{jkt}(t) v_{jk} v_{jk}^T
\]

(16)

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

Likewise, other types of SGMM acoustic model parameters such as \( v_{jk} \) and \( \Sigma_k \) 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 \( x_t \) given \( y_t \), 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, it is worthwhile to point out 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 \( x_t \), while the UBM is originally trained on the noise-corrupted features \( y_t \). Since the UBM provides the regression class for the Gaussian components when applying JUD [12], it needs to be in the same space as the SGMM. In this work, the UBM is updated using the weighted average of the corresponding Gaussian component in the SGMM as

\[
\Sigma_i^{\text{ubm}} = \Sigma_i
\]

(17)

\[
w_i^{\text{ubm}} = \frac{\sum_{jkt} \gamma_{jkt}(t)}{\sum_{jkt} \gamma_{jkt}(t)}
\]

(18)

\[
\mu_i^{\text{ubm}} = \frac{\sum_{jkt} \gamma_{jkt}(t) M_i v_{jk}}{\sum_{jkt} \gamma_{jkt}(t)}
\]

(19)

where \( w_i^{\text{ubm}}, \mu_i^{\text{ubm}} \) and \( \Sigma_i^{\text{ubm}} \) 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 \( M \) by the standard maximum likelihood training.

2. For each training utterance, initialise the noise model parameters for \( n_t \) and \( b_t \) in (4).

3. Re-estimate the noise model parameters given \( M \).

4. Obtain the JUD transformation parameters \( T \).

5. Given \( M \) and \( T \), compute the posterior probability \( \gamma_{jkt}(t) \) using (7).

6. Accumulate the statistics using the conditional expectations (12) and update \( M \).

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 SL-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.

<table>
<thead>
<tr>
<th>Methods</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clean model</td>
<td>52.2</td>
<td>58.2</td>
<td>50.7</td>
<td>72.1</td>
<td>59.9</td>
</tr>
<tr>
<td>+JUD</td>
<td>5.1</td>
<td>13.1</td>
<td>12.0</td>
<td>23.2</td>
<td>16.8</td>
</tr>
<tr>
<td>MST model</td>
<td>68.8</td>
<td>15.2</td>
<td>18.6</td>
<td>32.3</td>
<td>22.2</td>
</tr>
<tr>
<td>+JUD</td>
<td>7.4</td>
<td>13.3</td>
<td>14.7</td>
<td>24.1</td>
<td>17.6</td>
</tr>
<tr>
<td>NAT model</td>
<td>6.5</td>
<td>20.3</td>
<td>19.8</td>
<td>39.7</td>
<td>27.6</td>
</tr>
<tr>
<td>+JUD</td>
<td>6.1</td>
<td>11.3</td>
<td>11.9</td>
<td>22.4</td>
<td>15.7</td>
</tr>
</tbody>
</table>

The value of phase factor $\alpha$ in the decoding stage.

Figure 1: Results of tuning the value of phase factor $\alpha$ in the decoding stage.

Acknowledgement This work is funded by the EPSRC Programme Grant EP/I031022/1 (Natural Speech Technology).
6. References


