Replacing Uncertainty Decoding with Subband Re-estimation for Large Vocabulary Speech Recognition in Noise

Jianhua Lu, Ji Ming, Roger Woods

The Institute of Electronics, Communications and Information Technology
Queen’s University Belfast
{jlu01,j.ming,r.woods}@qub.ac.uk

Abstract

In this paper, we propose a novel approach for parameterized model compensation for large-vocabulary speech recognition in noisy environments. The new compensation algorithm, termed CMLLR-SUBREST, combines the model-based uncertainty decoding (UD) with subspace distribution clustering hidden Markov modeling (SDCHMM), so that the UD-type compensation can be realized by re-estimating the models based on small amount of adaptation data. This avoids the estimation of the covariance biases, which is required in model-based UD and usually needs a numerical approach. The Aurora 4 corpus is used in the experiments. We have achieved 16.9% relative WER (word error rate) reduction over our previous missing-feature (MF) based decoding and 16.1% over the combination of Constrained MLLR compensation and MF decoding. The number of model parameters is reduced by two orders of magnitude.

Index Terms: noise robustness, noise compensation, missing-feature decoding, Aurora 4, speech recognition

1. Introduction

Robustness has been a critical part of modern speech recognition applications as it is known that the recognition accuracy degrades fast as the noise level increases. Parameterized model compensation has been proved to be effective for robust speech recognition, especially for stationary or slow-varying noises. Typical techniques include model-based feature enhancement (MBFE) [1], where the uncertainty with the front-end noise removal was dealt with at the back-end decoding stage: Vector Taylor series (VTS) [2, 3] and uncertainty decoding (UD) [4, 8] compensate the effect of noise on the models based on linearizing the relationship between the corrupted speech, clean speech and noise. Parameterized compensation often rises from a mathematical framework about the speech and noise, and maintains the same criterion for training and adaptation. Non-parameterized compensation such as missing-feature (MF) approaches [5] on the other hand, makes little assumption about the noise and avoids directly modeling the mismatch function. Instead, noise corrupted subband spectra are marginalized when calculating likelihoods. Our previous work [6, 7] used white noise at variable SNRs (signal-to-noise ratios) added to the training data to simulate real noise corruption. This is followed by MF decoding in the back-end which aims to reduce mismatched time-frequency components. This combination could significantly reduce the word error rates (WER) if noise spectra were properly covered by the scheme [7]. However, the model may lack desired sharpness to some specific noises.

This paper proposes a novel approach for parameterized model compensation, by combining UD with SDCHMM (sub-space distribution clustering hidden Markov modeling). The SDCHMM technique is employed to reduce the number of model parameters to be re-estimated, so that the UD-type compensation can be realized by re-estimating the models using computationally efficient EM algorithms based on small amount of adaptation data. This removes the need to use stereo data or numerical methods as often required for the UD-based compensation. The proposed new approach, termed CMLLR-SUBREST, is also similar to Constrained-MLLR speaker adaptive training (CMLLR-SAT) and aims to compensate for environmental variations based on adaptive data from the environment. The compensation can be implemented on the fly for large vocabulary speech recognition. The new approach can also be combined with the MF decoding [7] which generates the transcriptions of the data for unsupervised adaptation for improved accuracy. The new system has been evaluated on the Aurora 4 corpus for larger-vocabulary speech recognition in various noise conditions and has shown improved performance.

The organization of this paper is as follows. In Section 2, two compensation approaches, white noise retraining and UD, are studied in the paper. In Section 3, the proposed adaptation method CMLLR-SUBREST is presented. The results on Aurora 4 are shown in Section 4 and the paper is concluded in Section 5.

2. Model Compensation

In this section, two approaches to model compensation are reviewed: multi-style retaining with white noise corruption [7] and UD [4, 8]. The former is a non-parameterized approach, modeling noise effects from the time domain by adding white noise at various SNRs. The latter is a parameterized approach, directly modeling the mismatch function in the cepstral domain by piece-wise linear approximation. The two approaches will be utilized to form the proposed algorithm in the next section. Only additive noise is considered in the paper.

2.1. Retraining with White Noise Corruption

White noise retraining assumes that the effect of noise on speech can be modeled by adding white noise to different time-frequency regions at different SNRs. Assuming \( y \), \( x \) and \( n \) the corrupted speech cepstrum, clean speech cepstra and noise cepstra, respectively, their relationship can be approximated as follows:

\[
y(t, k) \approx f(x(t, k), n_{l_{i,k}}(t, k))
\]

where \( (t, k) \) are the indexes for time and frequency channel, and \( n_{l_{i,k}}(t, k) \) is a white noise cepstrum at SNR level \( l_{i,k} \) that has a similar form to the real noise cepstrum \( n(t, k) \). Therefore

\[
\]
instead of training the models directly from corrupted speech \( y \), which is ideal but may not be always practical, we train the models using speech corrupted by white noise at variable SNRs, i.e., the white noise compensated model which is given below

\[
p(x|s, \hat{M}) = \sum_{m=1}^{M} \omega_m p(x|s, m, \hat{M}) = \sum_{m=1}^{M} \omega_m N(x; \tilde{\mu}_sm, \tilde{\Sigma}sm)(2)
\]

where \( m, s \) are the indexes of mixture and state, respectively, and \( M \) is the model trained with speech data corrupted by white noise within a predefined SNR range. The advantages of the approach are that it avoids modeling the non-linear function \( f(x, n) \), and that no knowledge of noise is required except that the noise has a flat spectrum within each frequency channel. The MF technique may be introduced into this model to marginalize the mismatched components at the decoding stage. We have found that the WER reductions due to the white noise retraining and due to the MF decoding are additive [7].

While the above white noise retraining with MF decoding may account for a wide range of noise conditions, it lacks the desired sharpness to some specific noise conditions. In this paper, we combine it with UD by performing adaptation on the model \( M \), i.e., UD on top of the models trained with white noise corruption.

### 2.2. Uncertainty Decoding and CMLLR

Uncertainty decoding [4, 8] can be seen as a piece-wise linear compensation for noise. The likelihood of corrupted speech, \( y \), given state, \( s \), can be expressed as [4]

\[
p(y|s, M) = \int p(x|y|s, M)dx 
\approx \int p(y|x)p(x|s, M)dx (3)
\]

where \( M \) is the clean model trained with clean speech \( x \) only. If the conditional probability density \( p(y|x) \) is a GMM, then (3) can be parameterized as

\[
p(y|s, M) = \sum_{m=1}^{M} \omega_m |A_c|N(x; \mu_{sm}, \Sigma_{sm} + \Sigma_c)
\]

where \( c \) corresponds to the most likely noise class for front-end UD or the regression class associated with the mixture components for model-based UD. \( A_c, b_c, \Sigma_c \) are the linear transforms for the noisy cepstra, and \( \Sigma_c \) is a bias to the covariance matrix, accounting for the effect of noise.

UD is the extension of CMLLR with the bias term \( \Sigma_c \) added to the covariance. It can also be interpreted as a generalized/smoothed MF decoding algorithm [4]. The importance of the covariance bias is well explained in [8]. Note that if a global transform class for \( A_c, b_c, \Sigma_c \) is used, the determinant \( |A_c| \) will have no effect in the decoding and thus can be ignored. For simplicity, we assume the use of a global transform class in this paper.

UD often requires either stereo data to train a front-end model in the case of front-end UD which is not always available, or the use of a numerical method to estimate the covariance bias in the case of model-based UD. In the next section, we show that by using the SDCHMM technique to reduce the number of Gaussian parameters, we can avoid the estimation of the covariance bias but obtain the compensated covariance matrix as a whole through model retraining.

### 3. CMLLR-SUBREST Adaptation

#### 3.1. SDCHMM

Subspace distribution clustering (SDC) HMM was initially developed by Bocchieri [9] to reduce the number of Gaussian parameters and saves computations and memory. We use the SDCHMM principle to tie the model parameters so that they may be reliably re-estimated at subband level, using an efficient EM algorithm, with small amount of adaptation data. The aim, as described above, is to avoid the use of numerical methods as needed in the model-based UD algorithms.

A feature vector can be represented by a number of sub-streams concatenated, the distribution of which can be tied into a small compact Gaussian mixture pool with little loss of accuracy [9]. Assuming diagonal covariance, the likelihood of clean speech frame \( x \) of \( K \) sub-streams for state \( s \) can be expressed as

\[
p(x|s, M) = \sum_{m=1}^{M} \omega_m \prod_{k=1}^{K} N(x_k; \mu_{skm}, \Sigma_{skm})(5)
\]

This can be reduced to a compact form – the SDC form – by tying the sub-stream parameters across all states and mixtures:  

\[
p(x|s, M) \approx p^{dec}(x|s, M) = \sum_{m=1}^{M} \omega_m \prod_{k=1}^{K} N(x_k; \mu_{ik}, \Sigma_{ik})(6)
\]

where \( p^{dec}(x|s, M) \) stands for the SDCHMM with mean vectors \( \mu_{ik} \) and covariance matrices \( \Sigma_{ik} \), and

\[
S(s, m, k) : (\mu_{skm}, \Sigma_{skm}) \rightarrow (\mu_{ik}, \Sigma_{ik})(7)
\]

is the projection function mapping the original Gaussian component with index \( (s, m, k) \) to a codeword Gaussian in the compressed model space with index \( i, k \). The above projection reduces the original 3-dimensional model space – \( s \times m \times k \) – to a 2-dimensional space: \( i \times k \). The Bhattacharyya distance can be used for the projection from a continuous density HMM to an SDCHMM [9]. Note that the tying structure \( S(s, m, k) \) can be obtained either by clustering clean models or by clustering white noise trained models, as discussed in Section 2.1. In our experiments, the latter shows better results.

#### 3.2. SDCHMM-based Retraining as UD

Assuming a global transform class (and so the noise class subscript \( c \) can be ignored) and diagonal covariance matrices, we can express (4) into sub-streams in model space \( M \) (i.e., the model space trained with white noise corrupted data as defined in (2))

\[
p(y|s, \hat{M}) = \sum_{m=1}^{M} \omega_m |A| \prod_{k=1}^{K} N(z_k; \tilde{\mu}_{skm}, \tilde{\Sigma}_{skm} + \tilde{\Sigma}_k)(8)
\]

where \( z = Ax + b \) is the transformed feature vector and \( z_k \) is the \( k \)th sub-stream of \( z \).

Replacing the Gaussian components in (8) with the corresponding SDCHMM codewords as shown in (6), we obtain a SDCHMM version of the UD algorithm

\[
p^{dec}(y|s, \hat{M}) = \sum_{m=1}^{M} \omega_m |A| \prod_{k=1}^{K} N(z_k; \tilde{\mu}_{ik}, \tilde{\Sigma}_{ik} + \tilde{\Sigma}_k)(9)
\]
where $A$, $b$ and $\tilde{\Sigma}_k$ are the UD parameters to be estimated for noise compensation. In (9) for each sub-stream $k$, an overall covariance matrix can be defined by combining the codeword covariance and the unknown bias. Then we can rewrite (9) as follows:

$$p^{dec}(y|\lambda) = \sum_{m=1}^{M} \omega_{mk} |A| \prod_{k=1}^{K} N(\tilde{\mu}_{ik}, \tilde{\Sigma}_{ik})$$

where $\tilde{\Sigma}_{ik}$ represents the combined covariance matrix of the sub-stream codeword, i.e.,

$$\tilde{\Sigma}_{ik} = \Sigma_{ik} + \Sigma_k$$

Equation (10) has the form in which all the parameters, the transform $A$, $b$, and the covariance $\tilde{\Sigma}_k$, can be re-estimated using the forward-backward algorithm given the adaptation data. The mean $\tilde{\mu}_{ik}$ can also be included in the re-estimation. This makes the approach similar to CMLLR-SAT. The re-estimation is made feasible because of the small number of codewords due to the SDCHMM compression. The following shows the adaptation for both mean and covariance:

$$\tilde{\mu}_{ik} = \frac{\sum_{t} \gamma(i,k) x_{ik}}{\sum_{t} \gamma(i,k)}$$

$$\tilde{\Sigma}_{ik} = \frac{\sum_{t} \gamma(i,k)(x_{ik} - \tilde{\mu}_{ik})(x_{ik} - \tilde{\mu}_{ik})'}{\sum_{t} \gamma(i,k)}$$

where

$$\gamma(i,k) = \sum_{(i,k)=\mathcal{Z}(s,m)} \gamma(s,m)$$

and $\gamma(s,m)$ the forward/backward occupation probability of mixture $m$ of state $s$. Using (12), (13), we term Eq. (10) the CMLLR-SUBREST algorithm. In our paper, a codeword is re-estimated when $\sum_{t} \gamma(i,k)$ is above a predefined threshold. From the experiments described in section 4, we found that one utterance could cause up to 30 out of the 1920 sub-stream codewords used in the system to be re-estimated, and 10 utterances could be sufficient for retraining 90% of the codewords. Since not all codewords can be re-estimated given limited adaptation data, the MF decoding is used to reduce the contribution of those un-updated codewords to the recognition. The CMLLR part of the adaptation, i.e., the adaptation in (10), can be interpreted as a way of modeling the correlation between the sub-streams. The importance of the cross-stream correlation will be examined in the next section.

The adaptation procedure is summarized as follows:

1. Set $A = I$ and $b = 0$ and accumulate statistics $\gamma(s,m)$ and $\gamma(i,k)$ for CMLLR and parameter re-estimation, respectively.
2. For each codeword, if the accumulated statistics $\sum_{t} \gamma(i,k)$ > a predefined threshold, update the parameters with $(A,b)$; otherwise, skip to the next codeword.
3. Update $A,b$ using standard CMLLR with the updated parameters ($\tilde{\mu}_{ik}, \tilde{\Sigma}_{ik}$).
4. Go back to 2 until a local optimum reached.

Note that the iteration above consumes considerable amount of time mainly because of re-accumulating statistics for step 3 when $(\tilde{\mu}_{ik}, \tilde{\Sigma}_{ik})$ are updated. Nevertheless, the experiments show that three iterations provide significant WER reduction.

4. Experimental Results

The Aurora 4 database is used for the experiments. The database provides with two types of microphones, wv1 and wv2, and 6 types of real noises: car, babble, restaurant, street, airport and train. In this paper we only use the 16 kHz and wv1 test sets. For the training data, 7138 clean utterances are provided. We have first created a clean context-dependent recognition system consisting of 3241 states with 16 mixtures in each state. This is the original CDHMM. Then we transform the CDHMM to SDCHMM.

For implementing the MF decoding, we use subband features instead of MFCC. The subband features are derived from the de-correlated log filter-band energies (FBE) [10] with 20 mel-scaled filters. The average over the full 20 filter energies is appended to the subband feature vector. Delta and delta-delta coefficients of the static features are also added to the frame vector and treated as additional subband features. The dimension of feature vectors is 60. Each two neighboring components are grouped to form a subband. Deviations and accelerations are treated same as static features when forming subbands. Finally, the total number of sub-streams/subbands is 30. The corresponding SDCHMM has the same number of logical mixtures but only 64 codewords for each sub-stream. Following recommendations from the Aurora 4 database, 166 out of 330 utterances of the ARPA test set are used in the experiments, the 166 utterances consists of 8 speakers with about 20 utterances for each speaker. All adaptations are conducted in an incremental fashion. Several baseline systems with or without adaptation are compared.

4.1. Performance of Baseline Systems

We present three baseline systems to verify the proposed algorithm. The first system is quoted from our previous [7], i.e., MF decoding on top of a white noise trained model (MF+HMM). The second system is MF decoding on top of a SDCHMM projected from the white noise trained HMM (MF+SDCHMM). The projection reduces the number of Gaussian components from 51,856 to 64 codewords per sub-stream. The last system is the second system with the addition of CMLLR adaptation. The results are shown in Table 1. Note that all results are obtained using the MF decoding described in [7].

From Table 1 we can see that using SDCHMM instead of CDHMM increases the absolute WER by 3.2% on average (from Eq. (5) to Eq. (6)), but huge parameter size reduction (reducing the number of Gaussians per sub-stream from 51,856 to 64). Increasing the number of codewords per sub-stream from 64 up to 256 does not give significantly better performance. With the addition of the global CMLLR, shown in the last column of the table, the WER is reduced to 21.7%, which is better than the CDHMM (21.9%) without SDC model compression.

4.2. Performance of the Proposed CMLLR-SUBREST Adaptation: with and without MF Decoding

Table 2 shows the performance of the proposed CMLLR-SUBREST algorithm, with and without the use of MF decoding in the adaptation. With the use of the MF technique to select the optimal subbands in the decoding, we expect to obtain more accurate transcriptions/alignments than the fullband decoding for the unsupervised adaptation, and hence better adaptation performance. This is evident in Table 2, which shows a reduction of the absolute WER by 4.7% with the use of the MF decoding. Comparing with Table 1, Table 2 indicates that the proposed...
Table 1: WER (%) of three baseline systems: a CDHMM with 51,856 Gaussian per sub-stream, a SDCHMM with 64 Gaussians per sub-stream and without/with CMLLR adaptation, all using MF decoding.

<table>
<thead>
<tr>
<th>Adaptation</th>
<th>MF+HMM</th>
<th>MF+SDCHMM</th>
<th>CMLLR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clean</td>
<td>14.5</td>
<td>15.8</td>
<td>12.7</td>
</tr>
<tr>
<td>Car</td>
<td>19.1</td>
<td>25.2</td>
<td>15.8</td>
</tr>
<tr>
<td>Babble</td>
<td>23.2</td>
<td>25.3</td>
<td>23.1</td>
</tr>
<tr>
<td>Restaurant</td>
<td>27.1</td>
<td>30.2</td>
<td>26.3</td>
</tr>
<tr>
<td>Street</td>
<td>24.6</td>
<td>27.1</td>
<td>25.6</td>
</tr>
<tr>
<td>Airport</td>
<td>20.3</td>
<td>23.1</td>
<td>21.0</td>
</tr>
<tr>
<td>Train</td>
<td>24.8</td>
<td>29.2</td>
<td>27.5</td>
</tr>
<tr>
<td>Average</td>
<td>21.9</td>
<td>25.1</td>
<td>21.7</td>
</tr>
</tbody>
</table>

Table 2: WER (%) of the proposed CMLLR-SUBREST adaptation on the SDCHMM, without and with MF decoding.

<table>
<thead>
<tr>
<th>Adaptation</th>
<th>CMLLR-SUBREST</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MF decoding No</td>
</tr>
<tr>
<td>Clean</td>
<td>14.3</td>
</tr>
<tr>
<td>Car</td>
<td>18.5</td>
</tr>
<tr>
<td>Babble</td>
<td>26.7</td>
</tr>
<tr>
<td>Restaurant</td>
<td>28.5</td>
</tr>
<tr>
<td>Street</td>
<td>28.0</td>
</tr>
<tr>
<td>Airport</td>
<td>26.6</td>
</tr>
<tr>
<td>Train</td>
<td>30.1</td>
</tr>
<tr>
<td>Average</td>
<td>24.6</td>
</tr>
</tbody>
</table>

CMLLR-SUBREST, a re-estimation based implementation of UD, performed better than the CMLLR approach within our SDCHMM framework (19.9% in column 3 of Table 2 vs 21.7% in column 4 of Table 1).

4.3. Performance of the proposed CMLLR-SUBREST Adaptation: Effect of the Iteration

This section examines the effect of iterating the re-estimation of the CMLLR parameters \((A, b)\) given the SDCHMM parameters \((\tilde{\mu}_{ik}, \tilde{\Sigma}_{ik})\), and the SDCHMM \((\hat{\mu}_{ik}, \hat{\Sigma}_{ik})\) given the CMLLR \((A, b)\), as indicated in the CMLLR-SUBREST adaptation algorithm described in Section 3. Table 3 shows the results, indicating an increasing number of iterations improving the performance. After three iterations, the performance has become stable. Compared with Table 1 with the performance obtained by MF+HMM (i.e., 21.9%), the performance achieved by the new system (i.e., 18.2%) represents a 16.9% WER reduction, and at the same time with a reduction in the model size by more than two orders of magnitude.

5. Conclusions

In this paper, we presented a new noise compensation approach to large vocabulary speech recognition with small amount of adaptation data. It is achieved by using the SDCHMM structure to tie the model parameters at a sub-stream level, thereby reducing the model parameters to be updated. The model adaptation thus can be realized by re-estimating the model consisting of only a small number of sub-stream codewords. We also demonstrated that by combining with MF decoding, the compensation can be conducted more accurately in an unsupervised fashion with improved recognition accuracy. Our experiments also show that adaptation from models trained with white noise corruption produces better results than from clean models.

6. Acknowledgments

This work is supported by the UK EPSRC under Grant No. EP/D048605/1.

7. References