Noise Compensation for Speech Recognition Using Subspace Gaussian Mixture Models

Mohamed Bouallegue[1], Mickael Rouvier[2], Driss Matrouf[1], Georges Linarès[1]

[1]University of Avignon , LIA, France, [2]University of Le Mans, LIUM, France
{mohamed.bouallegue, driss.matrouf, georges.linares}@univ-avignon.fr
mickael.rouvier@lium.univ-lemans.fr

Abstract

In this paper, we adress the problem of additive noise which degrades substantially the performances of speech recognition system. We propose a cepstral denoising based on the Subspace Gaussian Mixture Models paradigm (SGMM). The acoustic space is modeled by using a UBM-GMM. Each phoneme is modeled by a GMM derived from the UBM. The concatenation of the means of a given GMM leads to a very high dimention space, called the supervector space. The SGMM paradigm allows to model the additive noise as an additive component located in a subspace of low dimension (with respect to the supervector space). For each speech segment, this additive noise component is estimated in a model space. From this estimation, a specific frame transformation is obtained and applied to such a data frame. In this work, training data are assumed to be clean, so the cleaning process is applied only on test data. The proposed approach is tested on data recorded in a noisy environment and also on artificially noised data. With this approach we obtain, on data recorded in a noisy environment, a relative WER reduction of 15%.

Index Terms: Acoustic modeling, Noise compensation, Subspace Gaussian Mixture Models, robust speech recognition.

1. Introduction

Automatic Speech Recognition (ASR) performance degrades substantially in the presence of ambient acoustic noise. This is mainly attributed to the mismatch between clean acoustic models and noisy test features. Usually, there are two ways to reduce this mismatch and achieve a satisfactory performance. One approach involves adapting back-end acoustic models according to the noisy environment. To apply this approach, it is necessary to estimate a background noise model and use a mismatch function that represents the impact of the noise on the clean speech. This mismatch function will allow to combine the estimated noise model, and the clean speech model to order to obtain a model for a noisy speech. In this category, we find the Parallel Model Combination (PMC) [1], Vector Taylor Series (VTS) [2] and Joint Uncertainty Decoding [3].

Another type of approaches is to denoise front-end feature vectors while keeping the clean models unchanged or develop robust noise features. To achieve it, a classical method is spectral subtraction [4]. It is a commonly used method for noise suppression where the additive noise spectrum is estimated and subtracted from the noisy speech spectrum to recover the clean speech spectrum. In [5], a Signal-to-Noise Ratio (SNR)-dependent cepstral normalization (SDCN) algorithm is proposed to compensate noisy speech features in the cepstral domain by removing the compensation vectors from them in a discrete HMM recognizer. The compensation vectors clustered by frame SNRs are estimated by using ”stereo” data which consist in clean and noisy speech signals recorded simultaneously [6].

In addition, in speech enhancement techniques, Signal Subspace Filtering class has gained a lot of attention because of its important role in ASR, improving the robustness in noisy environments [7]. In this approach, a nonparametric linear estimate of the unknown clean speech signal is obtained, based on a decomposition of the noisy signal into a signal subspace and an orthogonal noise subspace. The signal subspace contains vectors of the clean signal and the orthogonal subspace contains vectors of the noise process only. The noise reduction is obtained by nulling the noise subspace and removing the noise contribution in the signal subspace. A detailed theoretical analysis of the underlying principles of subspace filtering can be found, for example, in [8].

In this work, we will exploit the idea of projecting a vector representing a noisy speech in a subspace assumed to only contain the noise part. Hence, this projected component can be subtracted from the noisy vector to obtain a clean speech vector. This process is performed in the cepstral domain, assuming that the whole cepstral space is modeled by a Gaussian Mixture Model, called Universal Background Model (UBM). Each phoneme can be modeled by a GMM derived from the UBM using, for example, a MAP adaptation of the UBM vector means. The concatenation of the GMM means obtained leads to a vector with a very high dimension. We call this high dimension vector, a supervector. The projection in the noise subspace is performed in the supervector space. The results obtained show that at least a part of the noise can be located in subspace of low dimensional. This approach, called Subspace Gaussian Mixture Model (SGMM), has already been proposed in [9] for HMM-state modeling. In this original paper [9] the low dimension subspace contains the specific state information rather than additive noise. We also used this approach, in a previous works [10][11], for different applications for speech recognition.

This paper is organized as follows: in the next section, we explain how to model the additive noise effect within the SGMM paradigm. In Section 3, we present the strategy to estimate the noise component at once in the subspace of low dimension. Results are reported and commented in Section 4. Finally, we conclude and we present some perspectives in Section 5.

2. Subspace Gaussian Mixture Models For Additive Noise Modeling

As explained in the introduction, we assume that all cepstral vectors can be generated by a GMM, called here UBM. The UBM is defined as follows: UBM=$(\ _g, m_g, \ _g)$, where $\ _g$, m_g and $\ _g$ are respectively the weight, the mean and the covariance matrix of the g^{th} Gaussian. The GMM associated to a given phoneme ph is derived from the UBM. As introduced before, we call a supervector the concatenation of all GMM means. Note that m_{ph} is the supervector corresponding to the phoneme ph. In the presence of additive noise n, the supervector m_{ph} contains two components: a phoneme component and a noise component. If we assume that the noise component can be located in a subspace of low dimension (with respect to the supervector dimension) then we can write:

$$m_{ph,n} = m + Dy_{ph} + Ux_n \qquad (1)$$

The columns of the U matrix are the generative vectors of the noise subspace. U is a $MN \times R$ matrix, where N is the dimension of the cepstral vectors, M is the number of Gaussians in the UBM, and R is the chosen dimension of the noise subspace. x_n is the noise vector in the noise subspace generated by the columns of U, and the coordinates of x_n are called the noise factors. The vector y_{ph} models the linguistic content of the phoneme ph.

To estimate the parameters of the model in equation 1, we have to do some assumptions: both y_{ph} and x_n are assumed to be normally distributed among N(0; I). D is a diagonal matrix (a $MN \times MN$ matrix) so that DD^t is the *a priori* covariance matrix of the linguistic information component.

To estimate the U matrix, we need to noise each phoneme by using different types of noise: (ph, n) will denote the data frames of the phoneme ph corrupted by the noise n. We use different recordings of the noise collected in different environments such as bars, restaurants, and hospital. The noises are of same type, namely the sound of confused voices obtained from a crowd, but they are different from each other by changing the recording conditions (location, acoustic conditions ...). The noise is then added in a time domain. The set (ph, n) is obtained by grouping together the frames of each phoneme ph in the training corpus noised by the recording of the noise n.

The model of Equation 1 is not used for speech recognition, but only for noise component estimation. The noise estimation is then subtracted from the data frames and used for training or testing an ASR.

2.1. Estimation of noise components

In this paragraph, we detail the strategy to estimate the noise component in the low dimension U matrix. It is iteratively estimated using the Expectation Maximization (EM) algorithm. At each step, $x_{ph,n}$ is estimated, then y_{ph} is estimated for each phoneme (using the new x) and, finally, U is estimated globally, based on these $x_{ph,n}$ and y_{ph}. Since $x_{ph,n}$ and y_{ph} also depend on U, the process is iterated. The various individual steps are described in the algorithm 1. We describe next the equation required to estimate the U matrix.

Let $\mathbf{N}_{(ph)}$ and $\mathbf{N}_{(ph,n)}$ be two vectors containing respectively the zero order phoneme-dependent statistics and noise-dependent statistics . $\mathbf{X}_{(ph)}$ and $\mathbf{X}_{(ph,n)}$ two vectors containing the first order statistics. These statics are estimated against the UBM:

$$\mathbf{N}_{(ph)}[g] = \underset{t\quad ph}{\ }\ \ _g(t);\ \mathbf{N}_{(ph,n)}[g] = \underset{t\quad ph}{\ }\ \ _g(t) \qquad (2)$$

$$\{\mathbf{X}_{(ph)}\}_{[g]} = \underset{t\quad (ph)}{\ }\ \ _g(t)\cdot t;\ \{\mathbf{X}_{(ph,n)}\}_{[g]} = \underset{t\quad (ph,n)}{\ }\ \ _g(t)\cdot t \qquad (3)$$

where $\ _g(t)$ is the *a posteriori* probability of Gaussian g for the observation t. In the equation, $\ _{t\quad ph}$ represents the sum over all frames belonging to the phoneme ph.

Let $\overline{\mathbf{X}}_{(ph)}$ and $\overline{\mathbf{X}}_{(ph,n)}$ be the state dependent statistics defined as follows:

$$\begin{aligned}\{\overline{\mathbf{X}}_{(ph)}\}_{[g]} &= \{\mathbf{X}_{(ph)}\}_{[g]} - \underset{n\quad ph}{\ }\ \mathbf{N}_{(ph,n)}[g] \cdot \{\mathbf{Ux}_{(ph,n)}\}_{[g]} \\ \{\overline{\mathbf{X}}_{(ph,n)}\}_{[g]} &= \{\mathbf{X}_{(ph,n)}\}_{[g]} - \{\mathbf{m} + \mathbf{Dy}_{(ph)}\}_{[g]} \cdot \underset{ph}{\ }\ \mathbf{N}_{(ph,n)}[g]\end{aligned} \qquad (4)$$

Let $\mathbf{L}_{(ph,n)}$ be a $R \times R$ matrix, and $\mathbf{B}_{(ph,n)}$ a vector

```
Algorithm 1: Estimated algorithm of $U$
  For each phoneme $ph$ and noise $n$: $\mathbf{y}_{(ph)} \quad 0$,
  $\mathbf{x}_{(ph,n)} \quad 0, \mathbf{U} \quad random$ ;
  Estimate statistics: $\mathbf{N}_{(n)}$,$\mathbf{N}_{(ph,n)}$ , $\mathbf{X}_{(ph)}$, $\mathbf{X}_{(ph,n)}$
  (eq.2 and 3);
  for $i = 1$ to $nb\_iterations$ do
      for all $ph$ and $n$ do
          Center statistics: $\overline{\mathbf{X}}_{(ph)}$, $\overline{\mathbf{X}}_{(ph,n)}$ (eq.4);
          Estimate $\mathbf{L}_{(ph,n)}$ and $\mathbf{B}_{(ph,n)}$ (eq.5);
          Estimate $\mathbf{x}_{(ph,n)}$ and $\mathbf{y}_{(ph)}$ (eq.6);
      end
      Estimate matrix $U$ (eq. 7 and 8) ;
  end
```

of dimension R, both defined as:

$$\begin{aligned}\mathbf{L}_{(ph,n)} &= \mathbf{I} + \underset{g \ \text{UBM}}{} \quad \mathbf{N}_{(ph,n)}[g] \cdot \{\mathbf{U}\}^t_{[g]} \cdot \quad {}^{-1}_{[g]} \cdot \{\mathbf{U}\}_{[g]} \\ \mathbf{B}_{(ph,n)} &= \underset{g \ \text{UBM}}{} \quad \{\mathbf{U}\}^t_{[g]} \cdot \quad {}^{-1}_{g} \cdot \{\overline{\mathbf{X}}_{(ph,n)}\}_{[g]}\end{aligned} \tag{5}$$

By using $\mathbf{L}_{(ph,n)}$, $\mathbf{B}_{(ph,n)}$, $x_{(ph,n)}$ and $y_{(ph)}$ can be obtained using the following equation:

$$\begin{aligned}\mathbf{x}_{(ph,n)} &= \mathbf{L}^{-1}_{(ph,n)} \cdot \mathbf{B}_{(ph,n)} \\ \{\mathbf{y}_{ph}\}_{[g]} &= \frac{}{\ + \mathbf{N}_{ph}[g]} \cdot \mathbf{D}_g \cdot \quad {}^{-1}_{g} \cdot \{\overline{\mathbf{X}}_{(ph)}\}_{[g]}\end{aligned} \tag{6}$$

where is the MAP relevance factor. The matrix U can be estimated line by line, with $\{\mathbf{U}\}^i_{[g]}$ being the i^{th} line of $\{\mathbf{U}\}_{[g]}$ then:

$$\mathbf{U}^i_{[g]} = \mathbf{LU}^{-1}_g \cdot \mathbf{RU}^i_g, \tag{7}$$

where $\mathbf{RU}^i_g$ and $\mathbf{LU}_g$ are given by:

$$\begin{aligned}\mathbf{LU}_g &= \underset{ph \quad n \quad ph}{} \quad \mathbf{L}^{-1}_{(ph,n)} + \mathbf{x}_{(ph,n)}\mathbf{x}^T_{(ph,n)} \cdot \mathbf{N}_{(ph,n)}[g] \\ \mathbf{RU}^i_g &= \underset{ph \quad n \quad ph}{} \quad \{\overline{\mathbf{X}}_{(ph,n)}\}^{[i]}_{[g]} \cdot \mathbf{x}_{(ph,n)}\end{aligned} \tag{8}$$

The algorithm 1 presents the method adopted to estimate the noise matrix with the above developments where the standard likelihood function can be used to assess the convergence.

3. Noise Compensation for Speech Recognition

In the test step, we use the noise component U to filter the data frames in the cepstral domaine. The same U matrix was used for all utterances in the testing corpus. Each utterance was normalized using the following equation:

$$\hat{t} = t - \underset{g=1}{\overset{M}{}} \quad {}_g(t) \cdot \{U \cdot x_{utterance}\}_{[g]} \tag{9}$$

where M is the number of Gaussian components in the UBM, ${}_g(t)$ is the *a posteriori* probability of Gaussian g given by the frame t. These probabilities are estimated by using the UBM. And $U \cdot x_{utterance}$ is the additive noise component estimated on the utterance recording by an iteration of Algorithm 1. It is a supervector with $M \times D$ components. $\{U \cdot x_{utterance}\}_{[g]}$ is the g^{th} D component bloc vector of $U \cdot x_{utterance}$.

4. System Description and Results

4.1. Speeral

For this test, we used the LIA broadcast news ASR system, SPEERAL [12]. This system is based on an A* decoder using state-dependent HMM for acoustic modeling. The baseline Language Model (LM) is a 67k word broadcast news 3-gram, estimated on 200M words from the French newspaper Le Monde and from the ESTER broadcast news training corpus of about 1M words. The system uses context-dependent models trained on the 90 hours of ESTER transcribed data. State tying is performed by a decision tree algorithm, using acoustic context related questions.

4.2. Results

4.2.1. Test of real-world noisy data

Firstly, we tested our filtering approach on data recorded in a noisy environment. These data have been recorded during several conferences in degraded acoustic conditions: poorly placed microphone, background noise This corpus represents a real case of adverse environment for speech recognition. Moreover, the vocabulary used in this corpus was very varied. The two distortion sources are a great challenge for speech recognition where there are high Word Error Rates (WER).

Table 1 shows the results in term of WER for different recordings. The second and third columns show the WER of respectively unnormalized testing data and testing data normalized using the Equation (9).

Station	un-normalized	normalized
congress 1 (2h)	59.11	51.22
congress 2 (1h)	63.11	54.63
congress 3 (2h)	48.41	44.60
congress 4 (2h30mn)	63.99	53.81
Total (7h30)	58.01	50.55

Table 1: *Word Error Rates in % for different record.*

The results obtained show a performance improvement of the filtering based on the U matrix that exceeds 13% relative. These results give a primary confirmation of our proposal. They show the possibility to locate the noise in a low dimension subspace in the cepstral domaine. It is also very important to indicate that we use the same U matrix to normalize the different recordings, and it is not necessary to collect data for all conditions related to the current target application conditions.

4.2.2. Test of artificially noisy data

Remember that the data on which we have estimated the U matrix component were artificially noised. To test the importance of the noise level, we estimated three matrix on artificially noised data with signal-to-noise ratio (SNR) of 8db, 1db and -8db, respectively. We used these three matrices (called $\mathbf{U}_{N1}$, $\mathbf{U}_{N2}$ and $\mathbf{U}_{N3}$) to normalize the testing data, which have also been artificially noised with a SNR of -8dB.

Table 2 reports the results obtained in term of WER on our test corpus artificially noised. The first column contains the radio station names from which the samples were collected. The second column shows the WER of the unnormalized testing data. In the last three columns, the WER of the testing data is normalized by $\mathbf{U}_{N1}$,$\mathbf{U}_{N2}$ and $\mathbf{U}_{N3}$, respectively.

Station	un-normalized	$\mathbf{U}_{N1}$	$\mathbf{U}_{N2}$	$\mathbf{U}_{N3}$
Inter (4h)	39.92	38.52	37.71	36.43
RFI (1h10mn)	54.39	52.86	52.04	50.88
tvme (1h)	46.78	44.81	43.84	42.89
africa(1h30mn)	44.11	41.82	41.65	39.75
Total (7h40)	45.90	44.03	43.18	42.06

Table 2: *Word Error Rates in % for different radio stations.*

The largest gain was obtained when testing data were normalized with the matrix U_{N3}3. This shows the importance of estimating the U matrix on the data that have a SNR close to that of the testing data.

5. Conclusion

In this work, we proposed a feature normalization framework based on the SGMM paradigm. We used this technique to estimate the additive noise in a low dimension subspace in order to remove it from acoustic data frames in the cepstral domaine. Our approach has been tested on data recorded in a noisy environment but also on artificially noised data. the results obtained with the proposed approach show that at least a part of the noise can be located in subspace of low dimensional. The noise components are estimated on data in which the noise is additive in the time domain. The consuming time in the normalization procedure is very short, around 10 minutes per hour. This advantage will allow tu use this approach in real-time applications where the acoustic condition is degraded.

6. References

[1] M. J. F. Gales, “An improved approach to the hidden markov model decomposition of speech and noise,” in *ICASSP*, vol. 1, San Francisco, 1992, pp. 233–236.

[2] A. Acero, L. Deng, and J. Zhang, “Hmm adaptation using vector taylor series for noisy speech recognition,” in *INTERSPEECH*, 2000, pp. 869–872.

[3] H. Xu and K. K. Chin, “Comparison of estimation techniques in joint uncertainty decoding for noise robust speech recognition,” in *INTERSPEECH*, 2009, pp. 2403–2406.

[4] S. Boll, “Suppression of acoustic noise in speech using spectral subtraction,” vol. ASSP-27, no. 2, 1979, pp. 113–120.

[5] A. Acero, “Acoustical and environmental robustness in automatic speech recognition,” 1992.

[6] X. Cui and A. Alwan, “Noise robust speech recognition using feature compensation based on polynomial regression of utterance snr,” vol. 13, no. 6, 2005, pp. 1161–1172.

[7] K. Hermus, P. Wambacq, and H. V. Hamme, “A review of signal subspace speech enhancement and its application to noise robust speech recognition,” *EURASIP*, vol. 2007, 2007.

[8] H. Lev-Ari and Y. Ephraim, “Extension of the signal subspace speech enhancement approach to colored noise,” vol. 10, no. 4, 2003, pp. 104–106.

[9] D. Povey, L. Burget, M. Agarwal, P. Akyazi, K. Feng, A. Ghoshal, O. Glembek, and N. K. G. and, “Subspace gaussian mixture models for speech recognition,” in *ICASSP*, 2010, pp. 4330–4333.

[10] M. Bouallegue, D. Matrouf, M. Rouvier, and G. Linares, “Subspace gaussian mixture models for vectorial hmm-states representation,” in *ASRU*, 2011.

[11] M. Bouallegue, D. Matrouf, and G. Linares, “A simplified subspace gaussian mixture to compact acoustic models for speech recognition,” in *ICASSP*, 2011, pp. 4896–4899.

[12] P. N. ans C, Fredouille, G. Linarès, D. Matrouf, J.-F. Bonastre, , and F. Béchet, “Lia’s french broadcast news transcription system,” in *Lectures by Masters in Speech Processing*, 2004.