

# Maximum Likelihood Normalization for Robust Speech Recognition

*Yiu-Pong LAI, Man-Hung SIU*

Department of Electrical and Electronic Engineering  
Hong Kong University of Science and Technology, Hong Kong

harry@ust.hk, eemsiu@ust.hk

## Abstract

It is well-known that additive and channel noise cause shift and scaling in MFCC features. Empirical normalization techniques to estimate and compensate for the effects, such as cepstral mean subtraction and variance normalization, have been shown to be useful. However, these empirical estimate may not be optimal. In this paper, we approach the problem from two directions, 1) use a more robust MFCC-based features that is less sensitive to additive and channel noise and 2) propose a maximum likelihood (ML) based approach to compensate the noise effect. In addition, we proposed the use of multi-class normalization in which different normalization factors can be applied to different phonetic units. The combination of the robust features and ML normalization is particularly useful for highly mis-matched condition in the Aurora 3 corpus resulting in a 15.8% relative improvement in the highly mis-matched case and a 10.4% relative improvement on average over the three conditions.

## 1. Introduction

In recent years Automatic Speech Recognition (ASR) has been used very successfully. However, the performance of ASR systems degrades in the presence of additive noise or changing channels. Significant effort is focused on making speech recognition robust including both frontend-based techniques and model-based techniques. In speech recognition, Mel-frequency Cepstral Coefficient (MFCC) is one of the most commonly used features. To make speech recognition more robust against noise, one approach involves modifying the MFCC generation process, such as in [1]. Another possibility is the smoothing of the cepstral coefficients across time. Because the motion of vocal tract is relatively slow, the spectral power of speech generally changes slowly. However, the power of noise can change at any rate and, thus, some portion of noise can be removed by extracting the low modulation frequencies in the cepstral trajectory, in effect, smoothing the cepstral coefficients across time. Yet another approach is to normalize the MFCC, by shifting the mean and scaling the dynamic range to compensate for the effect of noise. Because the effect of additive noise is not linear in the cepstral domain, estimating the correct compensation at each time instance is not trivial. Techniques such as cepstral mean subtraction and variance normalization [2], involve the use of a single estimate across an utterance, have shown to perform reasonably well.

In this paper we discuss the use of a modified MFCC generation with time smoothing as described in Section 2. Then, we propose a maximum likelihood cepstral normalization approach. Instead of using the empirical mean and variance to normalize the cepstral features, an ML estimate is used. Because variance normalization can be equivalently applied to the

model, the estimation of the ML normalization parameters can be viewed as a constrained adaptation problem. The advantage of the model-based view is that multiple normalization factors can be applied to different phonetic units. The details of the ML normalization is described in Section 3. Extensive experiments were performed on both the Aurora 2 and Aurora 3 corpus, and the results are summarized in Section 4. Because a lot of the development work was performed before the ETSI Advanced DSR standard front-end was available, some reported experimental results measuring incremental improvements were based on last year's baseline. Final results, a combination of the ML normalization and the standard front-end, are also reported. The paper is concluded in Section 5.

## 2. Robust Feature Extraction

While MFCC is one of the most commonly used features in speech recognition, it can be improved so that it is more robust to additive noise especially when the noise condition is not known. In this research, we considered two techniques, they were: a) smoothing of the cepstral time trajectory and b) incorporation of higher order MFCC.

In addition to these modifications, power spectrum coefficients in FFT were used instead of magnitude coefficients in the filter bank computation for MFCC generation as suggested in [1].

### 2.1. Frequency selection of cepstral trajectory

As speech is produced by changing the shape of vocal tract, the rate of speech power variation is limited implying that the short-time difference in spectral power should be small. In this sense, speech information is mainly carried by low modulation frequencies of the feature vector. It has been shown that the low modulation frequencies of the cepstral domain are more important than the high modulation frequencies in speech recognition [3]. As higher modulation frequencies are more susceptible to noise corruption, removing high modulation frequencies by band-pass filters, such as the RASTA filter, has been shown to improve the recognition performance under noisy environments [4]. It is observed that the low cut-off frequency of the band-pass filter is very small and can be modeled as a low-pass filter together with a mean removal filter. As the mean of the feature vector is considered in normalization techniques, a simple low-pass filter, averaging filter, was used in our work.

### 2.2. Incorporation of higher order coefficients of MFCC

In the standard MFCC, the higher order coefficients of DCT are truncated. These coefficients mainly represent the pitch and vocal cords characteristics and, thus, are not commonly used in speaker independent speech recognition. However, the total

power of these coefficients, a useful indication of vocal cord vibration, can help to distinguish between voiced and unvoiced speech. This can be especially useful in the noisy environment in which it is more difficult to distinguish between sounds based on the spectral shape. The 12<sup>th</sup> MFCC coefficient is replaced by the total power of 12<sup>th</sup> to 22<sup>nd</sup> coefficients in our work.

### 3. Feature Vector Normalization

One approach used to compensate for noise effects in cepstral features is cepstral normalization. The most common normalization technique is cepstral mean subtraction (CMS), a blind channel normalization technique that produces a zero mean feature vector over an utterance.

While CMS can remove linear time-invariant channel effects, it may not be as useful for removing additive noise because additive noise also changes the dynamic range of the cepstral coefficients. Variance normalization has been proposed to normalize the feature vector to ensure zero mean and unit variance [2]. Variance normalization provides a method to re-scale the feature vector in order to reduce the differences between the feature coefficients of noisy speech and those of clean speech.

Denote the features of a  $T$ -frame speech utterance as a  $T \times D$  feature matrix  $X$  and the normalized feature matrix  $\hat{X}$  is computed as

$$\hat{x}_{t,d} = \frac{x_{t,d} - \bar{x}_d}{\sigma_d}, \quad (1)$$

where  $\bar{x}_d$  and  $\sigma_d$  are the sample mean and sample variance of the  $d^{\text{th}}$  component across the utterance.

We can also rewrite variance normalization as an affine transformation of the feature vector at time  $t$ ,  $X_t = [x_{t,1}, \dots, x_{t,D}]^T$ , to normalized feature,  $\hat{X}_t$ . That is,

$$\hat{X}_t = A^{-1}(X_t - b) \quad (2)$$

where  $A = \text{diag}\{a_1, \dots, a_D\}$  is a  $D \times D$  dimensional diagonal transformation matrix and  $b = [b_1, \dots, b_D]^T$  is  $D$  dimensional vector. The diagonal elements of  $A$ ,  $a_d$  equals to  $\sigma_d$  and  $b_d = \bar{x}_d$ . We called the normalization using empirical mean and variance as "empirical normalization".

#### 3.1. Maximum likelihood normalization

As mentioned before, because the effect of additive noise is not linear, the empirical normalization is only an approximation of the real effect of the noise. In fact, the utterance statistics are not only affected by the background noise, or channel effects but also by the speech contents and the amount of a non-speech in an utterance.

Instead of using the empirical estimate, normalization factors can be estimated using other criteria to increase robustness, such as the maximum likelihood (ML). The objective of ML normalization is to find the normalization parameters so that the model will maximize the likelihood of a speech utterance.

#### 3.2. Normalization as constrained adaptation

Although we initially considered the normalization in the feature space, because the normalization is affine, it can also be viewed as a model transformation. One advantage of the model-based approach is that different normalization factors can be applied to different speech models.

Denote  $f(\hat{x})$  as the likelihood of the transformed data  $\hat{x}$  evaluated against the Gaussian model with mean  $m_i$  and covariance matrix  $W_i$ .  $f(\hat{x})$  can be written as

$$f(\hat{x}) = N(\hat{x}; m_i, W_i) \quad (3)$$

In the model space, this likelihood is equivalent to evaluating the original feature  $X$  using a model which is normalized by the same parameters  $[A, b]$  in the following manner.

$$f(x; A, b) = N(x; Am_i + b, AW_iA^T) \quad (4)$$

Model based ML normalization can be considered a special type of model adaptation in which the normalization parameters  $[A, b]$  are applied to the mean and covariance matrix. To find the parameters for ML normalization, the EM algorithm can be used to obtain the ML estimate of  $[A, b]$ . Digalakis [5] has shown that the Expectation step (E-step) of the EM algorithm involves the computation of the sufficient statistics for each Gaussian mixture  $i$  as follows

$$\bar{\mu}_i = \frac{1}{n_i} \sum_{t=1}^T \rho(s_t) \phi_{i,t} x_t \quad (5)$$

$$\bar{\Sigma}_i = \frac{1}{n_i} \sum_{t=1}^T \rho(s_t) \phi_{i,t} (x_t - \bar{\mu}_i)(x_t - \bar{\mu}_i)^T \quad (6)$$

$$n_i = \sum_{t=1}^T \rho(s_t) \phi_{i,t} \quad (7)$$

where  $\phi_{i,t}$  is the posterior probability of the unobserved mixture indexes  $\omega_i(t)$  in state  $s_t$  given the current normalization parameters  $[A_0, b_0]$ , i.e.

$$\phi_{i,t} = P(\omega_i(t) | A_0, b_0, x_t, s_t) \quad (8)$$

and  $\rho(s_t)$  is the posterior probability of state  $s_t$  at time  $t$  given the observation sequence  $X$  and the current HMM parameters  $\lambda_0$ , i.e.

$$\rho(s_t) = P(s_t | X, \lambda_0). \quad (9)$$

The state posterior probability  $\rho(s_t)$  can be found using either the forward-backward algorithm or the state alignment of the recognition output as an approximation. We will discuss the estimation of  $\rho(s_t)$  in Section 3.3.

The scaling of the MFCC vectors as shown in Equation (1) implies that  $A$  is a diagonal matrix. If covariance matrices of the speech models are also diagonal as is often the case in speech recognition, the maximization step (M-step) of the EM algorithm can be simplified by solving the following set of one-dimensional quadratic equations. Denotes  $A = \text{diag}\{a_1, \dots, a_D\}$  and  $b = [b_1, \dots, b_D]^T$ . For each dimension,

$$\begin{aligned} & \left( \sum_{i=1}^N n_i \right) a^2 - \left( \sum_{i=1}^N \frac{n_i}{w_i^2} \right) b^2 - \left( \sum_{i=1}^N \frac{n_i m_i}{w_i^2} \right) a b + \\ & \left( \sum_{i=1}^N \frac{n_i \bar{\mu}_i m_i}{w_i^2} \right) a + \left( 2 \sum_{i=1}^N \frac{n_i \bar{\mu}_i}{w_i^2} \right) b - \left( \sum_{i=1}^N n_i \frac{\bar{\mu}_i + \bar{\Sigma}_i^2}{w_i^2} \right) = 0 \end{aligned} \quad (10)$$

and

$$b = \left( \sum_{i=1}^N \frac{n_i \bar{\mu}_i}{w_i^2} \right) - a \left( \sum_{i=1}^N \frac{n_i m_i}{w_i^2} \right) / \sum_{i=1}^N \frac{n_i}{w_i^2} \quad (11)$$

where  $m_i$  and  $w_i^2$  are the corresponding elements of the mean vector and diagonal covariance matrix  $W_i$  of Gaussian mixture  $i$ ,  $N$  is the total number of Gaussians in the model and  $\bar{\Sigma}_i^2$  is the diagonal coefficient of the sufficient statistics in (6).

Compared to the traditional MLLR, because diagonal transformation matrix is used, smaller number of parameters are required to be estimated making it possible to be estimated using smaller amount of data (for example using only one utterance to estimate the transformation). Furthermore, the variance of the models are also transformed in the normalization described above while MLLR typically only transforms the mean vectors.

### 3.3. Estimation of state posterior probability

The normalization matrix  $A$  is estimated to maximize the likelihood of the observations. This likelihood depends on the state posterior probability  $\rho(s_t)$  and can be computed using the forward-backward (FB) algorithm and we call this the **FB approach**. If we assume that the likelihood is dominated by the best state sequence, then the observation likelihood on the Viterbi path can be used to simplify the computation and we call this the **Viterbi approach**. However, the states in the Viterbi path may be erroneous so it is useful to include some of the lowest competitors into the ML likelihood estimation. Then, the  $N$ -best hypothesis can be used. Instead of weighting the  $N$  hypotheses by their likelihoods, they are weighted equally because the likelihood of the best hypothesis generally dominates and we call this the **N-best approach**. Under the N-best approach, the state posterior probability of state  $s_t$  at time  $t$  becomes

$$\rho(s_t) = \frac{1}{N} \sum_{n=1}^N \zeta_{n,t}(s_t) \quad (12)$$

where  $\zeta_{n,t}(s_t) = 1$ , if  $s_t$  is at time  $t$  of the  $n^{\text{th}}$  best hypothesis sequence and  $\zeta_{n,t}(s_t) = 0$ , if it is not. All three approaches to estimate  $\rho(s_t)$  are evaluated in Section 4.

### 3.4. Multi-class normalization

The effect of noise on different speech sounds, such as vowel or non-vowel may be different. So, it is preferable to use different normalization parameters for particular type of sounds. However, it is also important to have enough data to accurately estimate the normalization parameters. To balance the model specification and limitations of the adaptation data, similar Gaussians can be tied into one class and share the same transformation, similar to what is used in speaker adaptation. To illustrate the feasibility of multi-class normalization, we used two normalization matrices: one was for non-speech sounds, including silence and the short pause models, and the other was for speech states. In multi-class normalization, an unique set of normalization parameters are estimated for each class as follows.

$$\bar{\mu}_i(c) = \frac{1}{n_i} \sum_{t=1}^T \sum_{i \in \gamma(c)} \rho(s_t) \phi_{i,t}(c) x_t \quad (13)$$

$$\bar{\Sigma}_i(c) = \frac{1}{n_i} \sum_{t=1}^T \sum_{i \in \gamma(c)} \rho(s_t) \phi_{i,t}(c) (x_t - \bar{\mu}_i) (x_t - \bar{\mu}_i)^T \quad (14)$$

$$n_i(c) = \sum_{t=1}^T \sum_{i \in \gamma(c)} \rho(s_t) \phi_{i,t}(c) \quad (15)$$

where  $\gamma(c)$  represents a set of Gaussians in class  $c$ .

$$\phi_{i,t}(c) = P(\omega_i(t) | A_0(c), b_0(c), x_t, s_t)$$

where  $A_0(c)$  and  $b_0(c)$  are normalization parameters for class  $c$ .

## 4. Experiments

Two sets of experiments were performed on the Aurora 2 and Aurora 3 databases: a) on a development set that uses the traditional MFCC features on the Aurora 2 database for both clean and multicondition training and b) on a evaluation set that uses the ETSI Advanced DSR standard front-end [6] (would be called the standard frontend from now on) for feature generation for the Aurora 3 corpus. Due to our limited computational resources, we only evaluated the standard front-end on Aurora 3. In this paper, we discuss the case in which only one utterance was used to estimate the normalization parameters  $[A, b]$  as described in Section 3.2.

### 4.1. Development using Aurora 2

In the development set, the training was performed using HTK 3.1 [7] with HMM complex back-end configuration [8] for both the clean and multicondition training. The goal of these experiments was to evaluate the effectiveness of the proposed robust features and ML normalizations. A series of experiments were performed including:

1. empirical normalization using MFCC generated using HTK,
2. addition of robust features as described in Section 2 to empirical normalization,
3. ML normalization using robust features in Experiment 2, based on the FB, Viterbi and N-best approaches as described in Section 3.3, and
4. robust features with 2-class (speech and non-speech) ML normalization.

The results of the development set experiments are summarized in Table 1 in which the first row is the baseline results as suggested in [8]. Using the empirical normalization, significant improvements are obtained as shown in row 2 for both the clean and multicondition training. When using the robust features (filtering the cepstral trajectory and using higher order cepstral coefficients) as shown in row 3, about a 9% relative improvement is obtained for both conditions. For the ML normalization, we tested three different approaches for estimating the state posterior probability as discussed in Section 3.3. These included the forward-backward sequence, Viterbi path and N-best sequence, and their results are tabulated in rows 4 to 6. All are better than the empirical normalization while the forward-backward is the best for the multicondition training, and the N-best alignment is the best for the clean training condition. Similar results are obtained for two-class ML normalization as shown in rows 7 to 9. The two-class normalization is particularly useful for the clean training under the N-best approach. We hypothesize that because the non-speech models in clean training had very weak energy, this was why the mis-match during the test was particularly large. Furthermore, this large energy difference may have caused large fluctuations in likelihood values between paths and made the Viterbi path more dominating. In the multicondition training, using the FB approach with a single set of normalization factor is sufficient.

### 4.2. Evaluation using Aurora 3

To ensure that our results were comparable with other researchers, we evaluated the results of Aurora 3 using the standard front-end for feature generation. Training again was based on the scripts provided with the corpus. The results of Aurora 3 are tabulated in Table 2. The first column tabulates the average results across all conditions and languages. The relative improvements, tabulated in the second column, were computed according to [9]. The first row is the reference baseline in [9]. Unfortunately, we were unable to reproduce this baseline exactly. Our reproduction of the baseline, shown in the second row, gives a 4% relative degradation. The exact reason for this difference is still under investigation. The reasons why such a difference occurred could be because of the version of HTK, machine precision or other experimental variations. The use of empirical normalization with robust features gave an relative improvement of 3% compared to [9] as shown in the third row. Because the Aurora 3 training is not clean, it is more similar to the multicondition training in Aurora 2 rather than to the clean

Table 1: Word Error Rate of Aurora 2 using traditional MFCC

Aurora 2 Word Error Rate		
	clean	multi
No normalization	41.94%	12.97%
Empirical	17.59%	7.83%
Robust Features	15.81%	7.22%
ML (FB)	15.70%	<b>7.01%</b>
ML (Viterbi)	15.72%	7.05%
ML (N-best, N=10)	15.55%	7.16%
2 class ML (FB)	15.36%	7.09%
2 class ML (Viterbi)	15.56%	7.95%
2 class ML (N-best, N=10)	<b>14.60%</b>	7.60%

Table 2: Recognition performance on Aurora 3.

Aurora 3 Performance		
	Word Error Rate	Rel Improvement
Ref. baseline	9.69%	0%
Our baseline	10.27%	-3.88%
Empirical	8.99%	3.16%
ML (FB)	8.37%	10.42%

condition. So, we used the FB approach with one-class ML normalization. The result, shown in the fourth row, shows a 10% relative improvement over the reference baseline.

#### 4.3. Effect on different SNR and matching conditions

In addition to the average improvement, it is interesting to see how the robust features and the ML normalization performed under different SNRs. Table 3, tabulated according to [8], shows the Aurora 2 results under different SNRs when using different robust techniques in the development set. The robust techniques are particularly useful in low SNR conditions. Similarly, in Table 4, the average recognition results of Aurora 3 and their relative improvement under different conditions are tabulated. Consistent with the observation in regard to Aurora 2, the proposed approaches are particularly useful for large corruptions, i.e. for highly mis-matched conditions. The result shows an improvement on average, but the results for the highly mis-matched condition are significantly better. It is even more significant if we consider the fact that our own replication of the baseline was already 4% worse than the reference baseline. In fact, when we used our own baseline as the reference, the proposed algorithms gave an impressive improvement of 13%.

## 5. Conclusion

In this paper we presented a normalization technique using maximum likelihood criterion in combination with a robust feature generation. The performance of this technique was evaluated on the Aurora 2 and Aurora 3 noisy digit databases. The ML normalization shows a performance improvement in both databases. Furthermore, the proposed approaches are particularly useful for highly corrupted speech such as in low SNR or highly mis-matched conditions between training and testing. More importantly, improvement can be ensured when applying the proposed techniques either with standard MFCC frontend or with a robust frontend built specifically for handling noisy environments.

Table 3: Word Error Rate of Aurora 2 multicondition training at different SNRs

Aurora 2 Word Error Rate					
	20 dB	15 dB	10 dB	5 dB	0 dB
No normalization	2.27%	3.28%	5.67%	13.61%	40.03%
Empirical	0.94%	1.55%	3.11%	8.31%	25.44%
Robust features	0.99%	1.48%	2.99%	7.74%	22.89%
ML (FB)	0.96%	1.43%	2.88%	7.44%	22.33%

Table 4: Summary of recognition performance on Aurora 3

Aurora 3 Reference Word Error Rate					
	Finnish	Spanish	German	Danish	Average
Well (x40%)	3.91%	3.36%	4.89%	6.63%	4.70%
Mid (x35%)	19.08%	6.08%	9.16%	18.51%	13.21%
High (x25%)	13.39%	8.45%	8.75%	20.41%	12.75%
Overall	11.59%	5.58%	7.35%	14.23%	9.69%

Aurora 3 Word Error Rate					
	Finnish	Spanish	German	Danish	Average
Well (x40%)	4.29%	3.07%	3.93%	6.35%	4.41%
Mid (x35%)	11.90%	4.97%	10.32%	17.66%	11.21%
High (x25%)	10.21%	6.80%	8.14%	17.75%	10.73%
Overall	8.43%	4.67%	7.22%	13.16%	8.37%

Aurora 3 Relative Improvement					
	Finnish	Spanish	German	Danish	Average
Well (x40%)	-9.72%	8.63%	19.63%	4.22%	5.69%
Mid (x35%)	37.63%	18.26%	-12.66%	4.59%	11.95%
High (x25%)	23.75%	19.53%	6.97%	13.03%	15.82%
Overall	15.22%	14.72%	5.16%	6.55%	10.42%

## 6. References

- [1] Macho, D., Mauuary, L., Neo, B. Cheng, Y.M., Ealey, D., Jouviet, D., Kelleher, H., Pearce, D. and Saadoun, F., "Evaluation of a noise-robust DSR front-end on Aurora databases", Proc. ICSLP'02, 2002.
- [2] Viikki, O., Bye, D. and Laurila, K., "A recursive feature vector normalization approach for robust speech recognition in noise", Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing., p733-736, Seattle, USA, 1998
- [3] Kanedera, N., Arai, T., Hermansky, H. and Pavel, M. "On the importance of various modulation frequencies for speech recognition", Proc. European Conference on Speech Communication and Technology, 1997.
- [4] Hermansky, H. and Morgan, N., "RASTA Processing of Speech", IEEE Trans. Speech and Audio Proc., 2(4):587-589, 1994.
- [5] Digalakis, V.V., Rtschev, D. and Neumeyer, L.G., "Speaker adaptation using constrained estimation of Gaussian mixtures" IEEE Trans. Speech and Audio Proc., 3(5):357-366, 1995.
- [6] ETSI standard doc. "Speech processing, Transmission and Quality aspects (STQ); Distributed speech recognition; Advanced front-end feature extraction algorithm; Compression algorithms" ETSI ES 202 050 v1.1.1, October 2002.
- [7] Young, S., Evermann, G., Kershaw, Moore, G. D., Odell, J., Ollason, D., Valtchev, V. and Woodland, P. The HTK Book for HTK 3.1. Microsoft Corporation, Dec 2001.
- [8] [http://icslp2002.colorado.edu/special\\_sessions/aurora/](http://icslp2002.colorado.edu/special_sessions/aurora/)
- [9] [http://www.symporg.com/eurospeech/specialsessions/robustness\\_v2.html](http://www.symporg.com/eurospeech/specialsessions/robustness_v2.html)