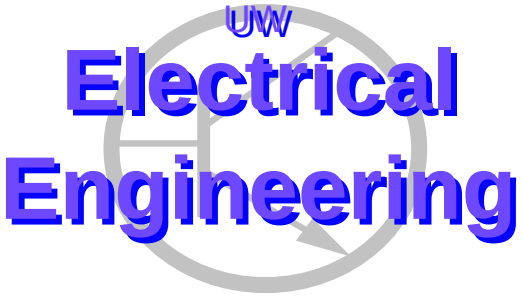
Towards Simple Methods of Noise Robustness

Chia-Ping Chen, Katrin Kirchhoff, Jeff Bilmes
{chiaping,katrin,bilmes}@ee.washington.edu

SSLI Lab
Dept of EE, University of Washington
Seattle, WA 98195-2500



UWEE Technical Report
Number UWEETR-2002-0002
January 2002

Department of Electrical Engineering
University of Washington
Box 352500
Seattle, Washington 98195-2500
PHN: (206) 543-2150
FAX: (206) 543-3842
URL: http://www.ee.washington.edu

Towards Simple Methods of Noise Robustness

Chia-Ping Chen, Katrin Kirchhoff, Jeff Bilmes
{chiaping,katrin,bilmes}@ee.washington.edu

SSLI Lab
Dept of EE, University of Washington
Seattle, WA 98195-2500

University of Washington, Dept. of EE, UWEETR-2002-0002
January 2002

Abstract

We introduce an effective and simple noise-robust feature processing technique which achieves very good results on the Aurora noisy-digits database. This technique does *not* require knowledge of the noise type and level. Also, it does not require any increase in modeling parameters. It performs well both on matched and mis-matched training and testing environments. In comparison to the Aurora baseline results, it improves relative performance by 45% in the case of multi-condition training and 60% in the case of clean training. The improvement is most profound in the noisiest of cases. Our feature processing technique can be easily integrated into other noise-robust feature processing schemes and noise-robust speech models to possibly yield further improvements. In other words, the simplicity of our technique suggests that it might be generally applicable.

1 Introduction

The performance of automatic speech recognition (ASR) systems is often poor when there is a mismatch between training and testing environments. Such a mismatch is often characterized either as additive (background) noise, or convolutional (channel) distortion, or both. The Aurora digits database[1] provides a nice framework for research on issues of noise-robustness. On this database, many different schemes have been tried to improve noise-robustness. In [2], an acoustic modeling procedure consisting both of a front-end neural network pre-processing step and Gaussian-Mixture models are used together. Principal component analysis (PCA) is further used to decorrelate features. In [3], missing-data theory and gender-dependent acoustic models are implemented. In [4], the dynamic SPLICE algorithm, which estimates the undistorted cepstrum given the observed cepstrum, is examined. This algorithm learns the biases in various environments and uses these biases to compensate noise within cepstra. In addition, blind equalization techniques are utilized to fight convolutional distortion. In [7], variable frame rate analysis, peak isolation, harmonic demodulation, and peak-to-valley ratio locking are used. In [5], voice activity detection, RASTA speech processing, artificial neural networks, linear discriminant analysis, mean and variance normalization, and PCA are all integrated and used together to achieve noise-robustness. In general, researchers have used quite complex and diverse sets of models and front-end processing schemes on this task, and as a result, performance has improved markedly on both matched and mis-matched training/testing environments.

In this paper, we propose a noise-robust feature post-processing methodology that is extremely simple but at the same time very effective. By inspecting in the cepstral domain time sequences of the same utterance in clean and noisy environments, we saw the need for a method to make them as similar as possible. If this could be achieved, the performance across different types and levels of noise might be similar to that of the clean case. Our experiments show that our approach achieves comparable performance to the currently best known technique [2][1], which is computationally much more expensive.

[1] On the Aurora 2.0 multi-condition training set.

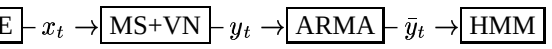


Figure 1: Block diagram of our feature processing method. FE: feature extraction, MS: mean subtraction, VN: variance normalization, ARMA: smoothing filter, HMM: ASR training and decoding

This paper is organized as follows. In section 2, we describe our feature processing technique. In section 3, we analyze our proposed method. In section 4, we present the experimental results and compare them with the results in [1]. In section 5, we comment on the results and draw conclusions.

2 processing in the feature space

We use standard mel-frequency cepstral coefficients (MFCC), $c_0 \dots c_{12}$, along with their deltas and double-deltas as our starting features. For each component of the feature vector, we perform the following steps.

Let N be the dimension of the feature space, and T be the number of frames in an utterance. Let x_{it} denote the i-th component of feature vector at frame t, where $i \in \{1, 2, \dots N\}$, and $t \in \{1, 2, \dots T\}$. Define

$$y_{it} = (x_{it} - \mu_i)/\sigma_i \tag{1}$$

where

$$\mu_i = \frac{1}{T}\sum_{t=1}^{T} x_{it} \tag{2}$$

$$\sigma_i = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(x_{it} - \mu_i)^2} \tag{3}$$

The above feature processing from x to y is well known to the ASR community as cepstral mean subtraction (MS) (subtracting μ_i) and element-wise variance normalization (dividing by σ_i).

After mean subtraction and variance normalization, we temporally process the features with a non-causal mixed Auto-Regression and Moving-Average (ARMA) low-pass filter on each component of the feature vector separately.

$$\bar{y}_{it} = \begin{cases} \frac{\sum_{k=1}^{M} \bar{y}_{i(t-k)} + \sum_{l=0}^{M} y_{i(t+l)}}{2M+1} & \text{if } M < t \le T - M, \\ y_{it} & \text{otherwise} \end{cases} \tag{4}$$

A block diagram of our post-processing technique is given in Figure 1. The feature processing methodology is performed per utterance both on the training and on the testing data. Note that we perform variance normalization and low-pass filtering in the *same* domain, namely both in the cepstral domain after MFCC processing has occurred. This is different from proposed approaches in the past that perform filtering (often in the spectral or log-spectral domain) and the variance normalization (often in the cepstral domain) in *different* domains [6, 5]. Note that switching the order of the variance normalization and the filtering in our method would not affect the processing technique (because they are both linear operations). It would be quite qualitatively different, however, to perform either of these processing steps before the final discrete-cosine transform (DCT) of MFCC processing.

3 the frequency response of the ARMA filter

In equation (4), it can be seen that the mean-subtracted and variance-normalized time sequences of all components are further processed with an ARMA filter. Interestingly, this operation results in significant improvements as will be demonstrated below. The idea of smoothing out a spiky time sequence is quite simple. While in clean speech certain spikes might contain important information about the speech utterance, in noisy speech many of the spikes are caused by noise. Therefore, when choosing the order M of the filter, there is an inherent trade-off. Choosing a small M will retain the short-time cepstral information, but is vulnerable to noise. Choosing a large M will be less susceptible to noise-caused spikes, but the short-time cepstral information will be lost. We therefore empirically evaluate a number of different M values for our results, as shown later in the paper.

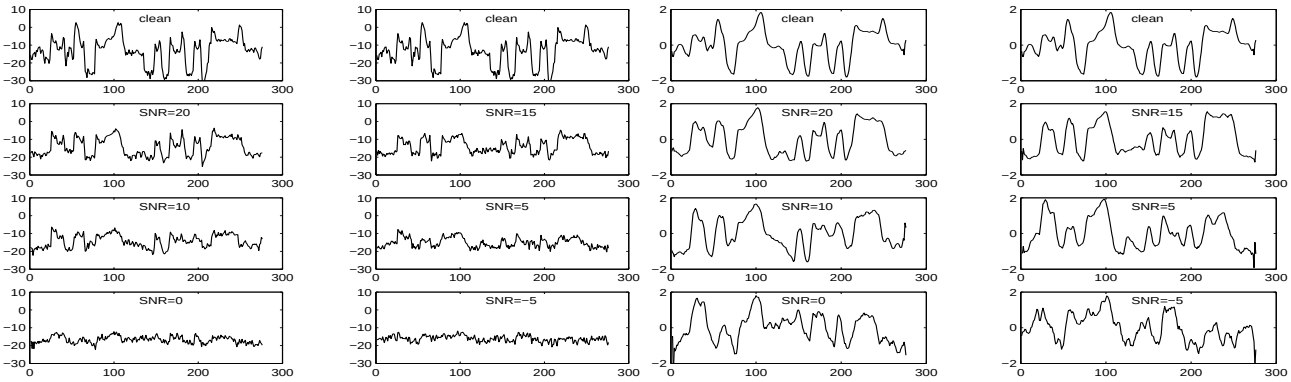


Figure 2: The time sequence of cepstral coefficient c_1 for the digit string "5376869" corrupted with different levels of noise before (left 8) and after (right 8) our proposed feature processing method. In this case, $M = 3$.

An illustration is given in Figure 2, which shows the time sequences of the same digit-string uttered in different noise levels. Note how the mean subtraction and variance normalization jointly bring utterances in different noise levels to the same reference level (via mean subtraction) and scale (via variance normalization) and how the low-pass ARMA filter smooths the sequences toward temporal similitude.

In order to examine the relationship between y and $\bar{y}$ in the frequency domain, we can rewrite equation (4) as:

$$(2M+1)\bar{y}_{it} - \bar{y}_{i(t-1)} - \cdots - \bar{y}_{i(t-M)} = y_{it} + \cdots + y_{i(t+M)} \tag{5}$$

From (5), the transfer function is:

$$H(z) = \frac{1 + z + \cdots + z^M}{2M + 1 - z^{-1} - \cdots - z^{-M}} \tag{6}$$

The frequency response is:

$$\begin{aligned} H(e^{j\omega}) &= \frac{1 + e^{j\omega} + \cdots + e^{jM\omega}}{2M + 1 - e^{-j\omega} - \cdots - e^{-jM\omega}} \\ &= \frac{1 - e^{j(M+1)\omega}}{2M + 2 - (2M+1)e^{j\omega} - e^{-jM\omega}} \end{aligned} \tag{7}$$

Note that for $\omega = 0$, $H(e^{j\omega}) = 1$. There are $\lfloor \frac{M+1}{2} \rfloor$ zeros in the interval $[0, \pi]$ equally spaced at

$$\omega = 2n\pi/(M+1), n = 1, 2, \ldots, \lfloor \frac{M+1}{2} \rfloor \tag{8}$$

As a result, one must be cautious about choosing large M since doing so could filter out important information in the speech. This conjecture was supported by our experiments, namely performance degrades when M gets large.

The frequency responses of the cases $M = 2, 4$ are plotted in Figure 3.

4 experimental results

The training and test sets of the Aurora database are defined in [1]. There are two sets of training data. The multi-condition training set consists of both clean and noisy speech, while the clean training set consists only of clean speech. There are three sets of test data. Test set A is composed of speech with conditions matched to the multi-condition training set, test set B is composed of speech with non-matched background noise, while test set C is composed of speech with partly matched background noise and non-matched convolutional noise.

In all experiments we report, we use a simple HMM-based system using whole-word models, 16 states per-word, 3 Gaussian components per state, uniform segmental k-means Gaussian initialization, and EM-training allowing for Gaussian component vanishing when component probabilities get small. During training, we use a 3-state silence model at the beginning and end of each utterance, and a forced single-state short-pause model. During testing, we

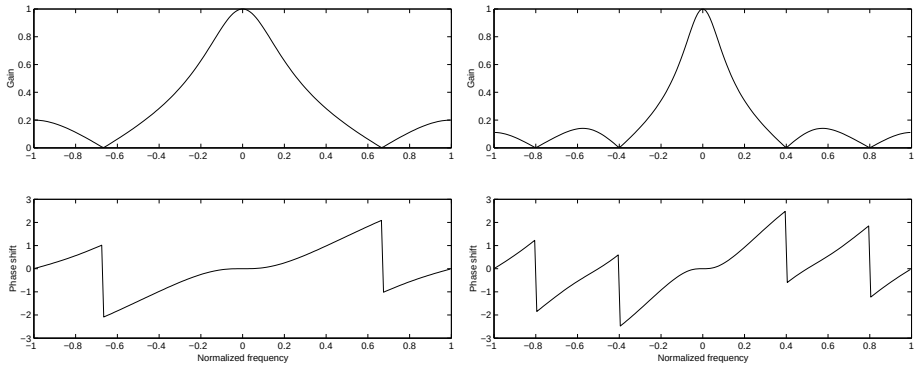


Figure 3: Gains and phase shifts of the ARMA filters of order M=2 (left 2), and 4 (right 2). They are low-pass filters with sidelobe peaks about 10 dB below that of the mainlobe.

Table 1: Percentage word accuracies of *noisy* (SNR = 0,5,10,15,20 dB) test speech under *multi-condition* training.

	Test A	Test B	Test C	Average	relative improvement
Baseline[1]	87.81	86.27	83.77	86.39	=
$M = 1$	92.21	92.53	91.89	92.27	43.2
$\boldsymbol{M = 2}$	**92.66**	**92.47**	**92.47**	**92.56**	**45.3**
$M = 3$	92.36	92.24	92.08	92.26	43.1
$M = 4$	92.18	92.05	92.01	92.09	41.9
$M = 10$	87.58	87.58	87.05	87.47	6.6

allow the silence model to optionally occur between words. In all our experiments, the total number of free parameters was no more than 42k.

We summarize our experimental results on noisy test speech (signal-to-noise ratios SNR=20, 15, 10, 5, 0 dB) under multi-condition training in Table 1 and under clean training in Table 2. We also summarize the results of the most noisy case (SNR=-5dB) in Tables 3 and 4. We also summarize the results of the clean case in Tables 5 and 6.

In the case of multi-condition training, $M = 2$ yields the best results (shown as bold in tables) out of all orders that were tried. This value apparently strikes a good balance between information preservation and noise robustness. In the case of only clean training, $M = 4$ (also shown in bold in the tables) yields the best results. Note the distinct behaviors of M in different training conditions. In multi-condition training, the performance achieves the optimum with relatively small M and then degrades fast afterward. With clean training, the performance achieves the optimum with medium M but degrades slowly thereafter. We include the case $M = 10$ to make this point clearer. We believe a larger M (i.e., filtering with more zeros) might become more beneficial in the mis-matched training-test conditions because of the greater disparity that exists in that case.

As can be seen, in both clean and noisy training conditions, there are significant improvements in noisy test conditions. It is remarkable that such simple post-processing of the MFCC features can yield such substantial improvements.

We also wanted to measure the effectiveness of just VN and ARMA filtering by themselves to better understand their relative merits. These experiments are summarized in Table 7 (multi-condition training, $M = 2$) and in Table 8

Table 2: Percentage word accuracies of *noisy* (SNR = 0,5,10,15,20 dB) test speech under *clean* training.

	Test A	Test B	Test C	Average	relative improvement
Baseline[1]	61.34	55.74	66.14	60.06	=
$M = 1$	81.16	82.59	80.97	81.69	54.2
$M = 2$	83.79	85.05	83.54	84.24	60.5
$M = 3$	83.73	84.82	82.87	83.99	59.9
$\boldsymbol{M = 4}$	**84.44**	**85.83**	**84.30**	**84.97**	**62.4**
$M = 10$	80.09	81.47	79.46	80.52	51.2

Table 3: Percentage word accuracies of *very noisy* (SNR = -5 dB) test speech under *multi-condition* training.

	Test A	Test B	Test C	Average	relative improvement
Baseline[1]	24.55	25.89	21.57	24.49	=
$M = 1$	44.04	43.79	44.30	43.99	25.8
$\boldsymbol{M = 2}$	**47.08**	**44.78**	**47.62**	**46.26**	**28.8**
$M = 3$	46.27	44.50	46.65	45.64	28.0
$M = 4$	46.34	44.83	46.91	45.85	28.3
$M = 10$	38.25	35.97	38.17	37.32	17.0

Table 4: Percentage word accuracies of *very noisy* (SNR = -5 dB) test speech under *clean* training.

	Test A	Test B	Test C	Average	relative improvement
Baseline[1]	7.93	7.65	11.48	8.52	=
$M = 1$	21.95	23.33	21.55	22.42	15.2
$M = 2$	28.08	28.79	27.30	28.21	21.5
$M = 3$	28.00	29.12	27.51	28.35	21.7
$\boldsymbol{M = 4}$	**28.50**	**29.26**	**28.55**	**28.81**	**22.2**
$M = 10$	27.91	27.98	27.17	27.79	21.1

Table 5: Percentage word accuracies of *clean* test speech under *multi-condition* training.

	Test A	Test B	Test C	Average	relative improvement
Baseline[1]	98.52	98.52	98.54	98.52	=
$\boldsymbol{M = 2}$	**98.80**	**98.80**	**98.51**	**98.74**	**14.9**

Table 6: Percentage word accuracies of *clean* test speech under *clean* training.

	Test A	Test B	Test C	Average	relative improvement
Baseline[1]	99.02	99.02	99.05	99.03	=
$\boldsymbol{M = 4}$	**99.18**	**99.18**	**99.04**	**99.15**	**12.4**

Table 7: Step-wise improvement of the proposed processing technique. Percentage word accuracies of *noisy* (SNR = 0,5,10,15,20 dB) test speech under *multi-condition* training. VN: variance normalization; ARMA: ARMA filter of order 2; VN + ARMA: VN followed by ARMA.

	Test A	Test B	Test C	Average	relative improvement
Baseline	87.81	86.27	83.77	86.39	=
VN	91.40	91.75	90.72	91.40	36.8
ARMA	89.76	90.81	90.39	90.31	28.8
VN+ARMA	92.66	92.44	92.47	92.53	45.1

Table 8: Step-wise improvement of the proposed processing technique. Percentage word accuracies of *noisy* (SNR = 0,5,10,15,20 dB) test speech under *clean* training.

	Test A	Test B	Test C	Average	relative improvement
Baseline	61.34	55.74	66.14	60.06	=
VN	77.64	79.34	77.28	78.25	45.5
ARMA	66.19	71.19	66.50	68.25	20.5
VN+ARMA	84.44	85.83	84.30	84.97	62.4

Table 9: Comparison of different low-pass filters. All the filters have $M = 2$. Shown in the table are the percentage word accuracies of *noisy* (SNR = 0,5,10,15,20 dB) test speech under *multi-condition* training.

	Test A	Test B	Test C	Average	relative improvement
non-causal ARMA	92.66	92.47	92.47	92.56	45.3
non-causal MA	92.36	92.42	92.12	92.34	43.7
causal ARMA	92.43	92.62	92.03	92.43	44.4
causal MA	92.08	92.38	91.92	92.17	40.6

Table 10: Comparison of different low-pass filters. All the filters have $M = 2$. Shown in the table are the percentage word accuracies of *noisy* (SNR = 0,5,10,15,20 dB) test speech under *clean* training.

	Test A	Test B	Test C	Average	relative improvement
non-causal ARMA	83.79	85.05	83.54	84.24	60.5
non-causal MA	84.02	85.75	84.85	84.88	62.1
causal ARMA	84.19	85.32	83.94	84.59	61.4
causal MA	81.45	82.71	81.23	81.91	54.7

(clean training, $M = 4$). Note that the CMS is always applied in these experiments. The results show that, while each technique applied individually significantly improves on the baseline, the two used together yield the best performance.

The next set of experiments show if there is an intrinsic advantage to ARMA processing, or if any low-pass filter suffices. We therefore ran our experiments with the following low-pass filters.

- causal ARMA filter
$$\bar{y}_{it} = \begin{cases} \frac{\sum_{k=1}^{M} \bar{y}_{i(t-k)} + \sum_{l=0}^{M} y_{i(t-l)}}{2M+1} & \text{if } M < t \leq T, \\ y_{it} & \text{otherwise} \end{cases}$$

- non-causal MA filter
$$\bar{y}_{it} = \begin{cases} \frac{\sum_{k=1}^{M} y_{i(t-k)} + \sum_{l=0}^{M} y_{i(t+l)}}{2M+1} & \text{if } M < t \leq T - M, \\ y_{it} & \text{otherwise} \end{cases}$$

- causal MA filter
$$\bar{y}_{it} = \begin{cases} \frac{\sum_{k=0}^{M} y_{i(t-k)}}{M+1} & \text{if } M < t \leq T, \\ y_{it} & \text{otherwise} \end{cases}$$

A comparison of our experimental results where we replace the non-causal ARMA filter used before with one of these filters is given in Table 9 (multi-condition training) and Table 10 (clean training). Note that because of the way we defined the causal MA filter, the results are given for a 3 (rather than 5) tap filter in that case. As can be seen, the difference in performances between causal/non-causal and ARMA/MA filters is small (the ARMA has a slight and probably insignificant advantage). This leads us to believe that, for this task, good results require only variance normalization and some form of low-pass filtering in the cepstral domain. The ARMA filter, however, has a particularly simple implementation (the array computation can be done entirely in-place).

5 Remarks and Conclusions

In this paper, we proposed an MFCC post-processing methodology which is both extremely simple but apparently extremely effective on the Aurora noisy digits database. It achieves the same level of performance as that achieved by much more sophisticated acoustic modeling and/or speech enhancements techniques. In general, our improvements are comparable to systems with many more free parameters, and which have significantly greater computational and conceptual complexity. Moreover, our method does not require knowledge about the noise, and performs well in both in matched and mis-matched acoustic environments between training and testing. Furthermore, the technique requires no additional free modeling parameters and does not require computationally more expensive training algorithms for parameter learning.

Our reported experimental results on clean training and noisy test conditions are of particular interests to us because of the implication to robustness in mis-matched acoustic environments. This is because clean speech is more environment-independent than is a particular noisy speech set — a technique where it is possible to train only in clean yet have good results in both clean and noisy speech is certainly desirable. To further investigate this issue, we plan further experiments using such environment-independent conditions.

Since our feature post-processing technique is performed within the feature space, there are no difficulties to combine our technique with other noise-robust acoustic models or noise-robust features. In future work, we will further experiment with variations of our method, both as a post-processing methodology for other forms of features, and for use together with more sophisticated statistical acoustic models. It remains to be seen if such integration can achieve further improvements.

6 acknowledgments

Many thanks to the members of SSLI Lab, and especially to Harriet Nock for many useful discussions. This work was funded by DARPA, Contract N660019928924.

References

[1] H. G. Hirsch and D. Pearce, “The AURORA Experimental Framework for the Performance Evaluations of Speech Recognition Systems under Noisy Conditions” ISCA ITRW ASR2000, Paris, September 2000.

[2] D. Ellis and M. Gomez, “Investigations into Tandem Acoustic Modeling for the Aurora Task”, Proceedings pp189-192, Eurospeech 2001.

[3] J. Barker, M. Cooke, and P. Green “Robust ASR Based on Clean Speech Models: An Evaluation of Missing Data Techniques for Connected Digit Recognition in Noise”, Proceedings pp213-216, Eurospeech 2001.

[4] J. Droppo, L. Deng and A. Acero, “Evaluation of the SPLICE Algorithm on the Aurora2 Database”, Proceedings pp217-220, Eurospeech 2001.

[5] C. Benitez, L. Burget, B. Chen, S. Dupont, H. Garudadri, H. Hermansky, P. Jain, S. Kajarekar, N. Morgan, S. Sivadas, “Robust ASR Front-end Using Spectral-based and Discriminant Features: Experiments on the Aurora Tasks”, Proceedings pp429-432, Eurospeech 2001.

[6] H. Hermansky, N. Morgan, “RASTA Processing of Speech” IEEE Transactions on Speech and Audio Processing, Vol. 2, No. 4, pp578-589, Oct. 1994.

[7] Q. Zhu, M. Iseli, X. Cui, A. Alwan, “Noise Robust Feature Extraction for ASR using the Aurora 2 Database” Proceedings pp185-188, Eurospeech 2001.