NOISE LEVEL NORMALIZATION AND REFERENCE ADAPTATION FOR ROBUST SPEECH RECOGNITION

Florian Hilger and Hermann Ney

Lehrstuhl für Informatik VI
RWTH Aachen – University of Technology
Ahornstr. 55
D – 52056 Aachen
{hilger, ney}@informatik.rwth-aachen.de

ABSTRACT

This paper describes an approach to normalize the noise level of a speech signal at the outputs of the Mel scaled filter–bank used in MFCC–feature extraction. An adaptive normalizing function that distinguishes between speech and silence parts of the signal is used to normalize the noise level, without altering the speech parts of the signal. This technique is combined with an adaptation of the reference vectors, depending on the average norm of the incoming feature vectors. On a database with training data recorded in office environment and testing data recorded in driving cars, the word error rate could be reduced from 35.5% to 14.7% for the city traffic testing set and from 78.0% to 24.1% for the highway testing set.

1. INTRODUCTION

Noise level normalization (NLN) is based on the observance, that a combination spectral subtraction (SS) and signal–to–noise–ratio normalization (SNRN) gives better recognition results when the subtraction and normalization are only applied to the noise parts of the signal and speech parts remain unchanged. Noise level normalization merges a normalization of the Mel frequency filter–bank outputs and the distinction between speech and silence in one normalizing function, that is multiplied to the filter outputs. Thus the number of parameters that have to be optimized empirically is smaller and training data can be used unaltered, which leads to a better recognition results on clean testing data.

2. SPECTRAL SUBTRACTION AND SNR–NORMALIZATION

In the context of this paper spectral subtraction denotes a subtraction after the Mel frequency filter–bank (figure 1). Here the noise estimate N_k for the k-th filter Y_k and its standard deviation σ_k are subtracted from the signal [6] , the clipping limit is a fraction ($\beta = 0.25$) of the mean noise estimate:

$$Y_k^{ss}[t] = \max(\,Y_k[t] - N_k[t] - \sigma_k[t]\,,\, \beta \cdot \frac{1}{K}\sum_k N_k[t]\,) \quad (1)$$

This setup gave better results than e.g. a constant overestimation factor and a constant clipping limit. After the spectral subtraction an SNR–normalization as described in [2] was carried out. A masking constant a_k is added to the signal, to keep the SNR at the target value, that was used in training.

$$Y_k^{snrn}[t] = Y_k^{ss}[t] + a_k[t] \quad (2)$$

a_k is increased, if the instant SNR is higher than the target SNR and decreased otherwise. The target SNR used for the experiments described below was 15dB.

Table 1 shows that the combination of SS and SNRN gives better recognition results than the baseline system for SNRs lower than the target SNR (see section 5 for database definitions and recognizer setup). For signals with higher SNRs the results are deteriorated.

Table 1: Recognition results on the TI digit string database with additional car noise at different SNRs. SS: Spectral subtraction, SNRN: SNR–normalization, NoSp: No SS and SNRN in speech parts, RefSw: Switch to clean references at low noise levels.

SNR [dB]	WER [%]			
	baseline	SS+SNRN	SS+SNRN NoSp	SS+SNRN NoSp RefSw
32	0.6	3.8	2.4	0.7
20	3.8	4.9	2.4	2.4
15	7.3	5.6	3.6	3.6
10	15.7	9.4	7.8	7.8
5	33.4	25.3	20.4	20.4
0	70.1	55.7	55.4	55.4

Comparing the output of the filters for clean and noisy signals, after taking the logarithm (figure 2), shows that the main mismatch lies in the silence parts of the signal. This effect can be exploited by applying SS and SNRN only to silence parts of the signal that lie below an energy threshold and leaving speech parts unchanged. This leads to an overall reduction of the error rate for all conditions. Except for the clean (32dB) condition, the error rate is always lower than the baseline. By adding a switch to change from SNR normalized references to the normal baseline references, if the noise level is low, the word error rate on clean data can be brought down to the the order of the baseline result.

It appears useful to apply a noise reduction technique that only changes the noise parts of the signal. The disadvantage of using SS and SNRN is the normalization of the training data to the target SNR which leads to a significant deterioration of the recognition performance on clean testing data, unless a reference switch is used. By using the noise level normalization described below, this problem can be avoided.

3. NOISE LEVEL NORMALIZATION

The noise level normalization approach normalizes the Mel frequency filter–bank outputs (figure 1) in one step by multiplying them with a normalizing function:

$$Y_k^{norm}[t] = H_k[t] \cdot Y_k[t] \tag{3}$$

The normalizing function includes an energy threshold that determines, if the current time frame is silence or speech. If silence is assumed, the filter output is scaled down to the fraction of the average speech level that was observed in training. Speech frames are left unchanged.

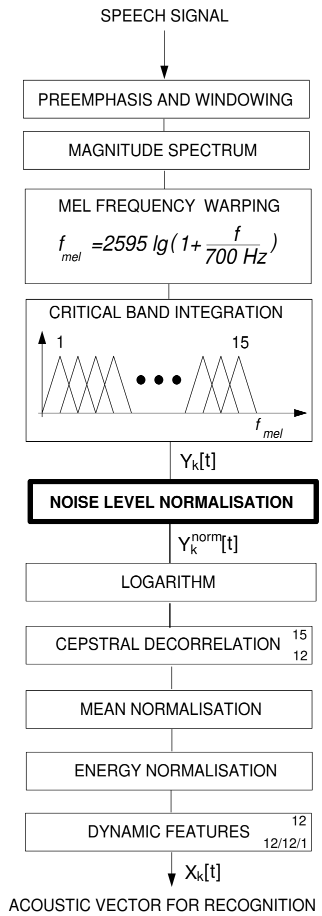

Figure 1: MFCC–feature extraction for a signal sampled at 8kHz with additional noise level normalization after the Mel scaled filter–bank.

Algorithm:

- Initialization: At the beginning of each utterance initialize the estimates of filter output levels for noise $\bar{Y}^n$ and speech parts $\bar{Y}^s$ of the signal. Calculate the initial noise level estimate form the first $T_{in} = 10$ to 20 time frames of the utterance:

$$\bar{Y}^n = \frac{1}{T_{in}K} \sum_t \sum_k Y_k[t] \tag{4}$$

Initialize the speech level estimate as multiple (e.g. $m = 2$) of the noise estimate:

$$\bar{Y}^s = m \cdot \bar{Y}_n \tag{5}$$

Fill two ring–buffers of $T = 50$ to 100 time frames length with these initial values.

- Update: While running over the sentence, update these buffers at every timeframe t. Calculate the average filter output magnitude $\bar{Y}$ for the time frame $t + T/2$:

$$\bar{Y}[t + T/2] = \frac{1}{K} \sum_k Y_k[t + T/2] \tag{6}$$

If it lies below a threshold (with $\beta = 0.2$)

$$\bar{Y}[t + T/2] < (1 - \beta)\bar{Y}^n[t] + \beta\bar{Y}^s[t] \tag{7}$$

consider the timeframe $t + T/2$ as noise t_n and update the noise buffer:

$$\bar{Y}^n[t] = \frac{1}{TK} \sum_t \sum_k Y_k[t_n] \tag{8}$$

Otherwise consider the timeframe as speech t_s and update the speech buffer

$$\bar{Y}^s[t] = \frac{1}{TK} \sum_t \sum_k Y_k[t_s] \tag{9}$$

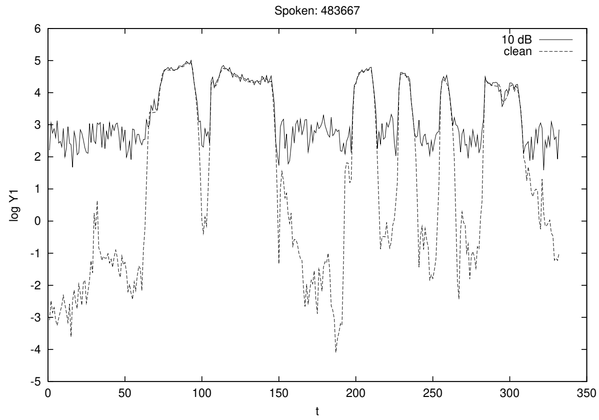

Figure 2: Example for the output of the first filter over the time (logarithmic scale). Clean signal and with additional car noise at 10dB.

In an offline system an approach to estimate the noise and speech levels by segmenting the utterance would presumably give better estimates, but the technique described here has the advantage that it is suitable for single pass online recognition, if a delay of $T/2$ (250ms to 500ms) is accepted.

- Training: In training run over the entire data (T_{tr}) to determine the average relation γ between the filter output levels in noise $\bar{Y}^n$ and speech parts $\bar{Y}^s$, without altering the signal:

$$\gamma = \frac{1}{T_{tr}} \sum_t \frac{\bar{Y}^n[t]}{\bar{Y}^s[t]} \tag{10}$$

  This constant γ is the fraction of the speech level to which the noise parts of the signal will be scaled down in the recognition.

- Recognition: During the recognition calculate a noise normalizing factor $\nu_k[t]$ for each filter output Y_k:

$$\nu_k[t] = \min\left(\gamma \cdot \frac{\bar{Y}^s[t]}{\bar{Y}_k^n[t]}\ ,\ 1\right) \tag{11}$$

$$\text{with} \quad \bar{Y}_k^n = \frac{1}{T} \sum_t Y_k[t_n] \tag{12}$$

  The noise level in recognition is calculated individually for each filter $\bar{Y}_k^n$, in order to normalize noise spectra that are not white. The speech level $\bar{Y}^s$ is still calculated globally like in equation (9) to get a more reliable estimation for short utterances. The normalizing factor is clipped at an upper limit of 1, to avoid unintended raising of the noise level.

- A sigmoid function of the average filter output magnitude i.e. the energy in the current time frame $\bar{Y}[t]$ (equation 6) is used as soft noise/speech threshold:

$$s(\bar{Y}[t]) = \frac{1}{1 + \exp\left(-\alpha\left(\frac{\bar{Y}[t]}{(1-\beta)\bar{Y}^n + \beta\bar{Y}^s} - 1\right)\right)} \tag{13}$$

  The parameters of the sigmoid function typically are $\alpha = 10$ and $\beta = 0.2$ (the same as in equation 7).

- Combining the normalizing factor ν_k and the sigmoid function s gives the actual normalizing function:

$$Y_k^{norm}[t] = \underbrace{(\nu_k[t] + (1 - \nu_k[t])s[t])}_{H_k[t]} \cdot Y_k[t] \tag{14}$$

  The slope of the normalizing function is shown in figure 3. A clean testing signal is not changed. Speech parts of noisy signals also remain unchanged $s[t] \approx 1 \Rightarrow H_k[t] \approx 1$, while the level of the noise parts is scaled down $s[t] \approx 0 \Rightarrow H_k[t] \approx \nu_k$.

4. REFERENCE ADAPTATION

A rising noise level pulls the MFCC feature vectors towards the cepstral vector of the noise. When using cepstral mean normalization this corresponds to a reduction of their norm. Their relative angles do not change so significantly.

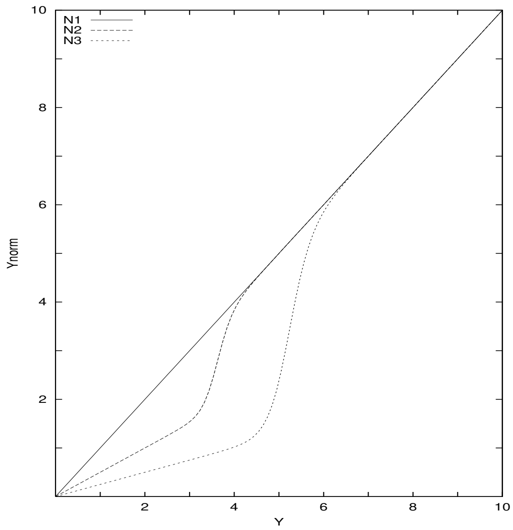

Figure 3: Mapping from Y_k to Y_k^{norm} at different noise levels, if $Y_k \sim \bar{Y}$ is assumed. N1: Clean (average noise level 1 arbitrary unit) no normalization, N2 : Noise level 2 units, N3 : Noise level 4 units

Rather than using alternative distance measures based on projection to compensate this effect as it is proposed in [4], the approach described here adapts the reference vectors in recognition, by means of a scaling factor depending on the average norm of the incoming feature vectors, while still using Laplacian or Gaussian log–likelihood calculation.

Algorithm:

- Calculate the average norm of the feature vectors $X[t]$ on the training data (T_{tr}):

$$l_{tr} = \frac{1}{T_{tr}} \sum_t \sqrt{||X[t]||^2} \tag{15}$$

- Determine l_{rec} for the utterance that is to be recognized. In the recognition tests described below, the entire (short) utterances where used to calculate l_{rec}. In an online application it could be calculated using a moving window.

- Calculate the normalizing factor:

$$n(l_{rec}) = \min\left(\frac{l_{rec}}{l_{tr}}, 1\right) \tag{16}$$

  or alternatively

$$n(l_{rec}) = \frac{1}{2} + \frac{1}{2}\cos\left(\pi + \min\left(\frac{l_{rec}}{l_{tr}}, 1\right)\pi\right) \tag{17}$$

- Multiply the references mean vectors with this factor when calculating the acoustic log–likelihoods:

$$-\ln p(\ X \mid n(l_{rec}) \cdot \mu\ ,\ \sigma\ ) \tag{18}$$

  If noise causes the average norm of the incoming feature vectors is smaller than in training, the reference vectors are scaled down proportionately. The covariances are not changed in this approach.

5. RECOGNITION RESULTS

Noise level normalization and reference adaptation where optimized on the TI digit string database with additional car noise and tested on an isolated word database with testing data recorded in driving cars.

TI digit strings with additional car noise

Database: The original TI digit string database [3] was sampled down from 16kHz to 8kHz to simulate the bandwidth of a telephone line. Except for this down–sampling the training data was left unchanged. A car noise sample was added to the clean testing data (SNR 32dB), to generate five noisy testing sets with lower SNRs.

Recognizer Setup:

- MFCC feature extraction (figure 1, [7])
  15 filters;
  12 cepstral coeff. + 12 first deriv. + 1 second deriv.
  = 25 – dimensional feature vector
- Word models for 11 English digits, including "oh"
- Gender dependent modeling
- Laplacian single densities
  - State dependent diagonal covariance matrix
  - No. of densities: $2 \cdot 358$

Table 2: Recognition results on the TI digit string database with additional car noise at different SNRs. NLN: Noise level normalization, RA: Reference adaptation.

TI Digit Strings + Car Noise				
SNR	WER [%]			
[dB]	baseline	NLN	RA	NLN+RA
32	0.6	0.6	0.7	0.7
20	3.8	2.3	2.7	1.9
15	7.3	3.9	4.3	2.6
10	15.7	8.0	8.3	4.1
5	33.4	18.8	18.8	9.5
0	70.1	40.1	46.2	25.9

Recognition Results: Compared to the baseline results the noise level normalization alone leads to an average relative reduction of the word error rates by 44.3% for the noisy conditions, without loss on the clean testing data (table 2). For spectral subtraction and SNR–normalization the average relative reduction was 39.5% (table 1). The computationally inexpensive reference adaptation alone performs almost as well. Combining NLN and RA leads to a further significant reduction in the word error rates. The average relative reduction is 64.6% compared to baseline.

With a recognizer setup similar to the one described above (word models for 10 English digits with $2 \cdot 337$ Laplacian single densities) additional tests where conducted on the NOISEX–92 database [5] (sampled down to 8kHz). The recognizer was trained on the clean training data and tested on data corrupted with the car noise sample provided with the database.

Table 3: Recognition results on the NOISEX–92 digit triplets with additional car noise at different SNRs. NLN: Noise level normalization, RA: Reference adaptation.

NOISEX–92 Digit Triplets + Car Noise				
SNR	WER [%]			
[dB]	baseline	NLN	RA	NLN+RA
18	0.3	0.3	0.3	0.3
12	1.0	0.7	0.3	0.3
6	4.3	2.7	2.0	0.3
0	52.7	37.0	16.6	5.7
-6	88.3	90.7	70.3	43.3

The results in table 3 show that the estimation of the noise and speech levels as it is described in section 3 does not work reliably any more on the testing set with negative SNR. The noise level normalization does not reduce the error rate, but nonetheless it has some effect on the signal that reduces the error rate significantly when it is used in combination with reference adaptation.

Isolated Word Car Navigation Database

Database: This German isolated word database was collected by RWTH for Panasonic Technologies Inc. / Speech Technology Laboratory. Training data was recorded in an office environment, with the microphone at 30cm distance directly in front of the speaker. The office testing set was recorded under the same conditions (average SNR 21dB). The other testing sets where recorded in driving cars (city and highway traffic, average SNRs 9dB resp. 6dB), with the speaker sitting in the passenger seat and the microphone mounted above the speaker on the visor. The sampling rate was 16kHz.

Table 4: German car navigation database. Each of the 12900 words occurring in the training data occurs only once and the 2100 words in the testing sets do not occur in the training data.

German Car Navigation Database				
	training	testing		
	office	office	city traffic	highway traffic
amount [h:min]	13:37	1:12	1:15	1:18
silence [%]	60			
speakers	86	14	14	14
spoken words	12900	2100	2100	2100
rec. vocabulary		2100 + 581 pron. variants		

Recognizer setup:

- MFCC feature extraction [8]
  20 filters;
  16 cepstral coeff. + 16 first deriv. + 1 second deriv.
  = 33 – dimensional feature vector
- Linear discriminant analysis [8]:
  Reduction from $3 \cdot 33$ components to 33
- Triphone classification and regression tree [1]:
  700 tied states

- Gender independent models
- Gaussian mixtures
  - Pooled diagonal covariance matrix
  - No. of densities: 20k – 21k
- Single word recognition

Table 5: Recognition results on the car database. NLN: Noise level normalization, RA: Reference adaptation.

German Car Navigation Database				
	WER [%]			
	baseline	NLN	RA	NLN+RA
office	3.8	4.2	3.9	4.3
city	35.5	21.1	26.1	14.7
highway	78.0	45.6	44.6	24.1

Table 6: Recognition results on the car database. The first two filters (centered on 89Hz resp. 190Hz) of the filter–bank where omitted.

German Car Navigation Database				
	WER [%]			
	baseline	NLN	RA	NLN+RA
office	4.5	5.1	5.3	5.4
city	28.9	16.8	22.2	14.4
highway	69.1	36.5	38.6	22.5

Recognition Results: The results (table 5) obtained on this database recorded in real car environment confirm the previous findings obtained on the TI digit string database: If NLN and RA are used alone the resulting reduction of the error rates on the noisy data is in the same order. Combining them leads to a further significant reduction of the word error rates. From 35.5% baseline to 14.7% for the city condition and from 78.0% to 24.1% for the highway condition. On clean data the error rate rises more significantly than on the TI digit string database. This can be explained by the short (one word) utterances that make an estimation of the speech and noise levels, less reliable and the more detailed acoustic modeling which is more sensitive to changes caused by normalization and adaptation.

The car noise usually has the largest effect on low frequencies (figure 4). By omitting the first (89Hz) an second (190Hz) filter of the filter–bank, the error rates can be further reduced on the car data (table 6). The cost is a rising error rate on the clean data.

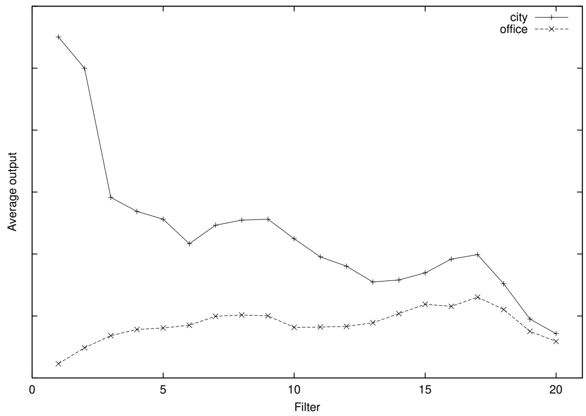

Figure 4: Average output of the filters for office and city traffic data. The noise has the largest effect on the first two filters.

6. CONCLUSIONS

This paper has presented an approach to increase the noise robustness of a speech recognizer by combining a noise normalization technique applied during feature extraction with an adaptation of the reference vectors.

The noise level normalization that was described scales the signals noise level in silence parts down to the level observed in training. In speech parts the mismatch caused by additional noise is smaller, so they remain unchanged. Reference adaptation uses a scaling factor multiplied to reference mean vectors, to compensate their decreasing norm in the presence of noise.

It was shown that both techniques used alone reduce the error rates on noisy data to a similar extent. Combining them leads to error rates significantly lower than the ones obtained with either of the approaches alone.

REFERENCES

[1] K. Beulen, E. Bransch and H. Ney, "State Tying for Context Dependent Phoneme Models," *Proc. 5th European Conf. on Speech, Communication and Technology,* Rhodes Greece, Vol. 3, pp. 1179–1182, September 1997.

[2] T. Claes and D. van Compernolle "SNR-Normalisation for Robust Speech Recognition," *Proc. 1996 IEEE Conf. on Acoustics Speech and Signal Processing,* Atlanta GA, pp. 331–334, May 1996.

[3] R. G. Leonard. "A database for speaker-independent digit recognition," *Int. Conf. on Acoustics, Speech and Signal Processing 1984*, San Diego CA, pp. 42.11.1-42.11.4, March 1984.

[4] D. Mansour and B. H. Juang, "A Family of Distortion Measures Based Upon Projection Operation for Robust Speech Recognition," *IEEE Transactions on Acoustics Speech and Signal Processing,* Vol. 37, No. 11, pp. 1659–1671, November 1988.

[5] A. Varga, H. J. M. Steenneken, M. Tomilson and D. Jones, "The NOISEX–92 study on the effect of additive noise on automatic speech recognition" *Documentation on the NOISEX–92 CD–ROMs,* June 1992.

[6] S. Vaseghi, "Advanced Signal Processing and Digital Noise Reduction," *John Wiley & Sons Ltd*, pp. 243–260, Chichester, 1996.

[7] L. Welling, H. Ney, A. Eiden, C. Forbrig, "Connected Digit Recognition Using Statistical Template Matching," *Proc. 5th European Conf. on Speech, Communication and Technology,* Madrid Spain, Vol. 2, pp. 1483–1486, September 1995.

[8] L. Welling, N. Haberland and H. Ney, "Acoustic Front End Optimization for Large Vocabulary Speech Recognition," *Proc. 5th European Conf. on Speech, Communication and Technology,* Rhodes Greece, Vol. 4, pp. 2099–2102, September 1997.