Noise-Robust Speech Recognition

Bernhard M. J. Leiner

Department of Computer Sience and Engineering
Helsinki University of Technology
bleiner@gmail.com

Abstract

The presence of noise in the real world is a major problem in the every day usage of speech recognition systems since it normally leads to a significant performance decrease. The main problem is the mismatch between training and operational data.

This paper tries to give an overview about the different techniques and algorithms used to improve accuracy for recognizing speech in a noisy environment. In practice ASR systems use a combination of several of this methods to achive better recognition rates.

In general we can seperate the techniques into two classes. The first one concentrates to on the compensation of noise during the preprocessing stage (feature mapping). This can be done by extracting features in a way that the feaure vectors are not affected by the noise. The other possibility is to actively reduce the noise by filtering or transforming the data. The second class of techniques tries to adapt the recognizer model to compensate the influence of noise. This paper briefly describes examples for all mentioned classes of techniques.

1. Introduction

In the past a lot of speech recognition systems were developed. One of the main problems that prevents a wide adoption of this technique in the everyday world is the problem of noise. Although the recognition rates are already close to 100 percent for some laboratory experiments the noise that can be present in the real world results in a major decrease of the recognition rate.

From a high level viewpoint the problem of noisy speech recognition is the mismatch between training and operating conditions. Figure 1 shows how much better the recognition rate is if the system was trained with noise data. To describe this mismatch, Gong [2] introduces the following transformation f: Let s be the model of a recognition unit, e.g. a phoneme or word, e be an environment, and $q_e(s)$ be some quantity defined on s in the environment e. A transformation f is a mapping of quantities between two environments α and β to minimize the mismatch.

$$q_\beta(s) = f(q_\alpha(s)) \tag{1}$$

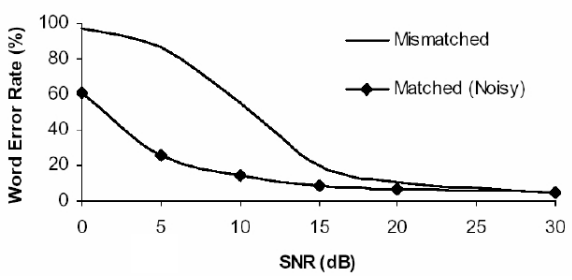

Figure 1: Word error rates for noisy and clean training data (from [1])

The straight forward approach would be to transform from the operating environment to the training environment (α = operating environment, β = training environment). Further discussion will be in section 2. But also the inverse transformation is possible and will be discussed in section 3.

The general problem with noise (and with finding the mentioned transformation function) is, that it is to a certain amount unpredictable. Car noise is different as noise in an office both in spectral characteristic and in expected loudness. In order to handle these different kinds of noise some constraints are made. For example noise is considered to be additive to the speech signal. Furthermore a lot of techniques to reduce the influence of noise are developed with the background that noise doesn't change as fast as the speech signals. To be able to compare different approaches to handle noise there exists noise databases (for example the NOISEX-92 database) with example noise like Lynx Helicopter noise or Operation Room noise.

One of the practical reasons noise is assumed to be additive is the ease of producing noisy training data by just adding noise to the clean speech.

Unfortunately there are additional effects that make the problem of noisy speech recognition harder. Firstly there is the so called *Lombard effect* [3]. It's caused by the fact, that humans articulation changes significant if they speak in a noisy environment. Secondly Openshaw and Mason [4] showed that additive gaussian noise results in several changes of the statistics of the cepstrum.

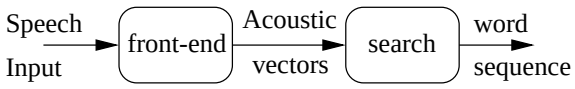

Figure 2: Speech recognition system

If we have a look at a speech recognition system as shown in figure 2 we can see two main modules. The front-end which is responsible for the feature extraction and the the recognizer. In order to gain noise robust recognition we can improve each of those two parts.

- Preprocess the data in such a way that the results looks like clean speech to the recognizer. This is also called "speech enhancement"
- Adapt the HMMs to match the noisy speech data. This can be done by switching to a HMM trained under the current kind of noise, by combining a clean speech model with a noise model to get a new HMM or by adapting parameters of the HMM.
- Combine both methods. Use preprocessing to decrease the SNR and adapt the HMM.

This paper [5] compares this three techniques for recognizing continuous speech in the presence of additive car noise.

2. Noise robust feature extraction

The aim of noise robust feature extraction is to deliver acoustic vectors to recognizer that are as close as possible to the training data. Close means that the distribution of the parameters is similar. Obviously impulsive noise results in a different distribution. But not only impulsive noise is a problem. As already mentioned, Openshaw and Mason [4] showed that also additive white noise also significantly alters the distribution. The mean shifts, the variance is reduced and the distribution becomes non-gaussian.

In principle there are two main possibilities to look at the problem of robust feature extraction:

- Detect the noise and transform the feature vectors into the training environment. (*speech enhancement*)
- Find a feature extraction method that gives the same results with and without noise. (*noise resistant features*)

The following sections will describe some example methods for robust feature extraction.

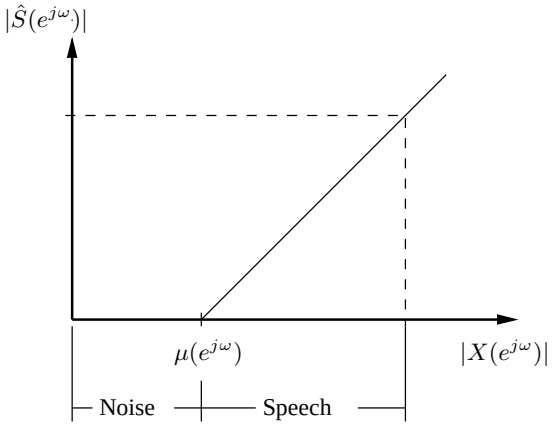

Figure 3: Input-output relation between $X(e^{j\omega})$ and $\hat{S}(e^{j\omega})$, [6]

2.1. Spectral subtraction

Spectral subtraction is a quite old speech enhancement method, introduced by Boll in 1979 [6]. The main advantage of this method is, that it's simple to understand and implement and computationally efficient.

Spectral subtraction uses an additive noise model. Furthermore noise and speech are assumed to be independent. The noisy speech signal $x(k)$ is the sum of the speech signal $s(k)$ and the noise signal $n(k)$[1]. This simple addition stays also after the Fourier transformation:

$$X(e^{j\omega}) = S(e^{j\omega}) + N(e^{j\omega}) \tag{2}$$

Since the noise level isn't exactly stationary the average $\mu(e^{j\omega})$ of the magnitude $|N((e^{j\omega})|$, taken during non-speech activity, is used. With this average and the phase $\theta_x(e^{j\omega})$ of $X(e^{j\omega})$ it is possible to calculate the *spectral subtraction estimator* $\hat{S}(e^{j\omega})$.

$$\hat{S}(e^{j\omega}) = (|X(e^{j\omega})| - \mu(e^{j\omega}))e^{j\theta_x(e^{j\omega})} \tag{3}$$

Equation 3 (figure 3 should clarify it) can also be written as

$$\hat{S}((e^{j\omega}) = H(e^{j\omega})X(e^{j\omega}) \tag{4}$$

where the filter $H(e^{j\omega})$ can be calculated as

$$H(e^{j\omega}) = 1 - \frac{\mu(e^{j\omega})}{|X(e^{j\omega})|} \tag{5}$$

The main problem is that the noise has to be stationary or only slowly varying ($\mu(e^{j\omega})$ can be adapted). Furthermore the decision, if there is speech or only noise, is really critical since mistakes would results in a major performance decrease.

[1] This linear connection isn't valid any more in the logarithmic spectrum domain

2.2. Minimum mean square error (MMSE) short-time spectral amplitude (STSA) estimator

This speech enhancement method was introduced in the year 1984 [7]. Like spectral subtraction the goal is to estimate the amplitude of the the clean speech signal given noisy speech data. This technique makes it possible to calculate the *optimal spectral amplitude* estimator in the ML sense whereas spectral subtraction STSA estimation is derived from optimal variance estimation.

The used model is once again additive noise in the time domain. The observed signal $y(t)$ is the sum of the speech signal $x(t)$ and the noise signal $d(t)$. Let X_k and Y_k denote the kth complex DFT coefficient of $x(t)$ and $y(t)$.

$$X_k = A_k \, e^{j\alpha_k} \tag{6}$$

The MMSE estimator $\hat{A}_k$ of A_k can be calculated like this:

$$\begin{aligned} \hat{A}_k &= E\{A_k \mid y(t), \qquad 0 \le t \le T\} \\ &= E\{A_k \mid Y_k\} \end{aligned} \tag{7}$$

The final result for the amplitude estimator is

$$\hat{A}_k = \frac{\sqrt{\pi}}{2} \frac{\sqrt{\nu_k}}{\gamma_k} e^{\frac{\nu_k}{2}} [(1+\nu_k) I_0(\frac{\nu_k}{2}) + \nu_k I_1(\frac{\nu_k}{2})] R_k \tag{8}$$

where ν_k and γ_k can be calculated out of the variances $\lambda_x(k)$ and $\lambda_d(k)$ of the kth spectral component of the speech and the noise.

2.3. Parameter mapping

Another speech enhancement method is parameter mapping. The goal is to map noisy speech observations into clean speech vectors with some kind of transformation. If it's possible to find such a transformation, like already introduced in equation 1, the accuracy of the ASR would be like without noise. Since we don't know this perfect transformation a number of approximated transformations were developed.

One example is a transformation that minimizes the square error. This prinzip is very easy and described for example in [8]. Let X and Y be the clean and noisy feature vectors. A linear transformation could be definded with a matrix A and a vector B.

$$Y \approx A \cdot X + B = Y' \tag{9}$$

The minimum square error E would be $|Y - Y'|^2$. With a given set of corresponding clean and noisy vectors the matrix A and the vector B can be calculated with the *Linea Multiple Regression* algorithm.

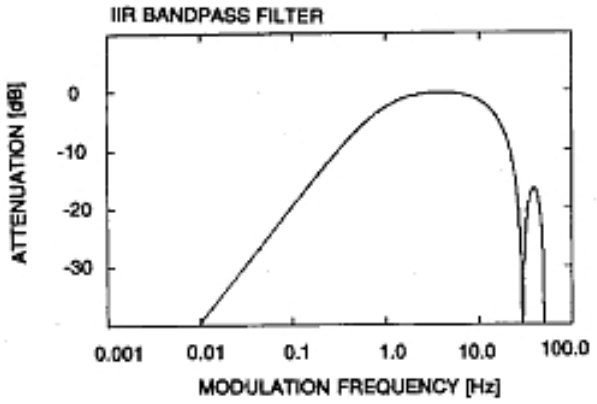

Figure 4: Frequency response of RASTA band-pass filter (from [9])

A different approach to do this transformation which gives better results would be to use an ANN. ANNs can be used for much more complex than just linear transformations. Sorensen investigated the usage of a multi-layer feedforward neural network model in his work [11].A major problem with this approach is that it is quite hard to train the network since clean speech and corresponding noisy noisey speech vectors are necessary.

2.4. RASTA processing

The relative spectral (RASTA) processing technique is an other example for a method to filter out noise and aims at noise resistant feature extraction. It is based on some basic characteristics of the human vocal tract and the human perception.

Linguistic messages are transformed into sound by movements of the vocal tract. Due to the physical characteristics of this vocal tract there are typical rates of change in the speech signal. The goal of RASTA processing, as described by Hermansky [9], is to suppress the spectral components that change more slowly or more quickly than the typical rate of change of speech.

Hermansky cites in this paper also early experiments that indicate the biggest sensitivity of human hearing at modulation frequencies around 4 Hz [2] It turns out that linguistic information is mainly modulated at frequencies between 1 Hz and 12 Hz.

Another motivaion for supressing slow varying audio data is the fact that steady background noise (e.g. in a cafeteria) doesn't serverly impair human speech communication. It looks like the human audio perception doesn't care much about slow varying audio data.

To simulate this in RASTA processing each frequency channel is filtered with a band-pass filter like in figure 4

[2] [10] is a more general paper that concentrates on the fundamental princips of human perception and how this knowledge can be used to build better ASR systems.

which filters unusual low and high modulation frequencies. The transfer function for this filter is[3]

$$H(z) = 0.1z^4 * \frac{2 + z^{-1} - z^{-3} - 2z^{-4}}{1 - 0.98z^{-1}} \quad (10)$$

RASTA processing is quite often combined with the PLP method which leads to the name RASTA-PLP. The first step of RASTA-PLP is to compute the critical-band power spectrum as in PLP. After that a compressing transformation is performed and the time trajectory of each spectral component is filtered. The next step is a exanding transformation and after that the conventional PLP processing resumes.

Note that there are different kinds of RASTA processing (e.g. J-RASTA) to handle both additive and convolutional noise.

2.5. Cepstrum mean normalization (CMN)

This technique is somehow similiar to RASTA processing since it also aims at the removal of slow variations. This is done by removing the long term mean from the the cepstral vectors [12]. Since this subtraction is done in the log-domain it reduces multiplicative disturbtions. Therefor it improves recognition accuracy for new channels (e.g. a different microphone).

An improvement of this classical CMN technique is *exact cepstrum mean normalization* (E-CMN) as described by Shozakai et al. [13]. It consists of two steps.

1. *Estimation step*: Two cepstrum mean vectors are calculated. One during speech which is speaker-dependent and the second one during periods without speech which is environment-dependent.

2. *Normalization step*: During speech the speaker-dependent vector is subtracted from the input speech frame and during non-speech the environment-dependent vector is used.

E-CMN results in a 2–3% improvement in the recognition word accuracy compared to CMN.

Since it is not possible to calculate the mean cepstrum vectors in advance for real-time application a running average is used if necessary.

3. Model compensation

Instead of trying to filter out the noise we can try to recognize speech although the data is polluted with noise. The possibility to train HMM with noisy data has the practical problem that the computational effort is rather high. Current techniques focus on training with clean speech and somehow adapting the recognizer.

[3]This filter is just an example used in [9] which was a bit improved later in this paper

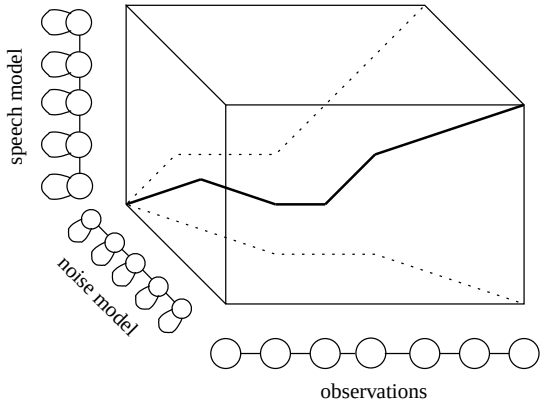

Figure 5: Decomposition of 3-dimensional state-sequence into two 2-dimensional projections in the noise and the speech state spaces; [14]

3.1. Parallel model combination

The basic idea of goes back to the work of Varga and Moore [14]. They proposed a general method of decomposing simultaneous processes that can be used in the problem of recognition contaminated with noise. This is done by using parallel hidden Markov models. One model for each of the components into which the signal is to be decomposed. An advantage of this technique is that in theory it's possible to use quite complex noise models to compensate also non stationary noise.

Figure 5 shows how this decomposition can be done for a signal consisting of speech and noise. We can see two 2-dimensional projections that shows the state of both the noise and the speech model (M_1 and M_2) for each observation. In general the probability of an observation can now be evaluated like this ($\otimes$ is some kind of combination operator):

$$\text{Observation Prob.} = P(\text{Observation} \mid M_1 \otimes M_2) \quad (11)$$

To find the most likely state sequence in a single HMM the *Viterbi* algorithm (equation 12) is used

$$P_t(i) = \max P_{t-1}(u) a_{u,i} b_i(O_t) \quad (12)$$

This algorithm can be adapted as in equation 13 to find the most likely sequence for a combination of two HMMs.

$$P_t(i,j) = \max P_{t-1}(u,v) a1_{u,i} a2_{v,j} b1_i \otimes b2_j(O_t) \quad (13)$$

$P_t(i,j)$ is the probability of the first component being in state i and the second in state j at time t. $a1$ and $a2$ are transition probabilities and $b1_i \otimes b2_j(O_t)$ is the observation probability.

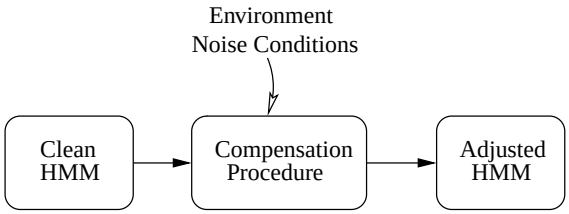

Figure 6: Creation of an adjusted model, [17]

Experiments showed (e.g. Gales and Young [15]) that with the usage of parallel model combination a performance level similar to training directly in the noise corrupted environment, can be achieved.

3.2. Model adaption

This techniques use a certain amount of test data to adapt the HMM model parameters to the noise environment. An overview is shown in Figure 6. In practice incremental adaption is used which means that the adaption data becomes available when the system is used.

A lot of research work in this area comes not from the problem of noise but from the problem of adapting speaker independent models into speaker adapted models. Proposed solutions for that can also be used for adapting models due to noise. This adaption can be done by a linear transformation. Gales compared in this paper [16] two possible forms of model based linear transformations. The basic equations look similar for both cases:

$$\hat{\mu} = A\mu + b \tag{14}$$

$$\hat{\Sigma} = H\Sigma H^T \tag{15}$$

where μ is the mean vector and Σ the covariance matrix. The difference between the compared methods is that one requires the variance transform to have the same form as the mean transform ($A = H$). This is a bit simpler and is described in detail in [18].

3.3. Noisy training data

In theory we would reach the highest recognition rates if the HMMs were trained under the same environment as the operational environment. In most cases this isn't practicable since it's hard to predict the noise and also the rest of the environment (e.g. used microphone). The second problem is the huge computational effort for training under all or at least most of the predicted operational conditions.

As a matter of fact this method for noise robust speech recognition is only suited for an environment that is quite restricted in terms of different kinds of noise. A car could be such an environment. Car manufacturers could use a defined microphone and the current speed and rotations per minute to choose the best matching HMM.

Recognition rates of systems trained with noisy training data are often used as benchmark for other noise robust speech recognizers.

3.4. Other approaches

Beside the few examples for quite classical methods for model compensation given in this paper there a plenty of others. Some would fit to a certain amount into the described categories some won't.

An interesting and recent approach is described in this paper [20] by Khan and Levinson. They use multiple HMMs (in general only 2 to reduce cost) the learn multiple views of the same input sequence. The advantage is that no adaption or training with noise takes place and hence no noise data is needed. The method is compared to view at asolid object from multiple angles.

Let X_f be a (forward) input sequence of speech features:

$$X_f = x_1, x_2, \ldots x_{n-1}, x_n \tag{16}$$

In order to get those multiple views the train the different the HMMs with different permutations of this input sequence. For example with X_f and the reverse sequence X_r which would be

$$x_r = x_n, x_{n-1}, \ldots x_2, x_1 \tag{17}$$

For recognition the input sequence and all permutations are evaluated with their according HMM. The output of all HMMs are evaluated based on the previous recognition trend, confidence factors, similarity, etc.

4. Conclusions

Although there was already done a huge amount of research effort the problem of noise in speech recognition is still not solved. The best results that are possible in practice come from a combination of several techniques. This starts with a suited microphone, a noise robust feature extraction, speech enhancement and model compensation. But even such a combinded method doesn't reach the recognition rates achieved with systems trained under the noisy conditions.

Another problem is that while some speech enhancement techniques are able to produce a speech signal that looks better on the spectrogram it isn't guaranteed that it really helps to improve the recognition accuracy.

If we don't combine methods but look a them seperatly, it's easier to obtain improved accuracy with some kind of transformation based techniques if the statistical properties of the noise are known. On the other side, with varying SNR noise robust feature extraction becomes more important.

Luckely the needed computational effort (which existst especially for model compensation) isn't a big prob-

lem any more due to modern DSPs in embedded systems or powerful PCs.

Another hope for better results is still the human auditory system. Humans are very good at filtering speech from noise. This is of course a result of the huge amount of reduntant information in natural languages and the context. But even if a person doesn't speak the spoken language and therefore doesn't understand a single word it is quite obvious for her which part of the audio signal belongs to speech and which is noise. As a matter of fact it looks like it is possible to distinguish sounds coming from the human vocal tract from noise even if there is no further information available.

5. References

[1] Pellom, B. "Speech Rcognition and Language Modeling: from Theory to Practice", Lecture slides, course T-61.184, Helsinki University of Technology, 2004 1

[2] Gong, Y. "Speech recognition in noisy environments: A survey", Speech Communication 16, pp. 261–291, 1995. 1

[3] Pisoni, D.B. et al. "Some acoustic-phonetic correlates of speech produced in noise", Proc. ICASSP, pp. 1581–1584, 1985 1

[4] Openshaw, J.P.; Mason, J.S. "On the limitations of cepstral features in noise", Proc. ICASSP, vol. 2, pp. 49–52, 1994 1, 2

[5] Neumeyer, L.; Weintraub, M. "Robust speech recognition in noise using adaption and mapping techniques", ICASSP-95,vol. 1, pp. 9–12, 1995 2

[6] Boll, S.F. "Supression of Acoustic Noise in Speech Using Spectral Subtraction", IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-27, pp. 113–120, 1979 2

[7] Ephraim, Y; Malah, D "Speech Enhancement Using Minimum Mean Square Error Short-Time Spectral Amplitude Estimator", IEEE Trans. On Acoustics, Speech, and Signal Processing, Vol. 32, No. 6, pp. 1109–1121, 1984 3

[8] Mokbel, C.; Chollet, G. "Word recognition in the car-speech enhancement/spectral transformations", International Conference on Acoustics, Speech, and Signal Processing, vol. 2, pp. 925–928, 1991 3

[9] Hermansky, H.; Morgan, N. "RASTA processing of speech" IEEE Transactions on Speech and Audio Processing, vol. 2, pp. 578–589, 1994 3, 4

[10] Hermansky, H. "Should Recognizers Have Ears?", Proc. ESCA Tutorial and Research Workshop on Robust Speech Recognition for Unknown Communication Channels, pp. 1–10, 1997 3

[11] Sorensen, H.B.D. "Noise-robust speech recognition using a cepstral noise reduction neural network architecture Neural Networks", IJCNN-91-Seattle, vol. 2, pp. 795–800, 1991 3

[12] Furui, S. "Cepstral analysis technique for automatic speaker verification" IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 29 , Issue: 2, pp. 254–272, 1981 4

[13] Shozakai, M.; Nakamura, S.; Shikano, K. "A speech enhancement approach E-CMN/CSS for speech recognition in car environments" IEEE Workshop on Automatic Speech Recognition and Understanding, Proceedings., pp. 450–457, 1997 4

[14] Varga, A.P.; Moore, R.K. "Hidden Markov model decomposition of speech and noise", Proc. ICASSP, pp. 845–848, 1990 4

[15] Gales, M.J.F.; Young, S.J. "Robust Continuous Speech Recognition Using Parallel Model Combination", IEEE Transactions on Speech and Audio Processing, vol. 4, pp. 352–359, 1996 5

[16] Gales, M.J.F. "Maximum-likelihood linear transformation for HMM-based speech recognition", Computer Speech and Language, 12, pp. 75–98, 1998 5

[17] Sanches, I. "Noise-Compensated Hidden Markov Models", IEEE Transactions on Speech and Audio Processing, vol. 8, pp. 533–540, 2000 5

[18] Digalakis, V.V.; Neumeyer, L.G. "Speaker adaptation using combined transformation and Bayesian methods", IEEE Transactions on Speech and Audio Processing, vol 4, pp. 294–300, 1996 5

[19] Matrouf, D.; Gauvain, J.-L. "Model compensation for noises in training and test data", ICASSP-97, vol. 2, pp. 831–834, 1997

[20] Khan, E.; Levinson, R. "Robust Speech Recognition Using a Noise Rejection Approach", International Journal on Artificial Intelligence Tools, vol. 8, no. 1, pp. 53–71, 1999 5