A New Method for Speech Denoising and Robust Speech Recognition Using Probabilistic Models for Clean Speech and for Noise

Hagai Attias, Li Deng, Alex Acero, John C. Platt

Microsoft Research USA
1 Microsoft Way, Redmond, WA 98052
{hagaia,deng,alexac,jplatt}@microsoft.com

Abstract

We present a new method for speech denoising and robust speech recognition. Using the framework of probabilistic models allows us to integrate detailed speech models and models of realistic non-stationary noise signals in a principled manner. The framework transforms the denoising problem into a problem of Bayes-optimal signal estimation, producing minimum mean square error estimators of desired features of clean speech from noisy data. We describe a fast and efficient implementation of an algorithm that computes these estimators. The effectiveness of this algorithm is demonstrated in robust speech recognition experiments, using the Wall Street Journal speech corpus and Microsoft Whisper large-vocabulary continuous speech recognizer. Results show significantly lower word error rates than those under noisy-matched condition. In particular, when the denoising algorithm is applied to the noisy training data and subsequently the recognizer is retrained, very low error rates are obtained.

1. Introduction

Denoising and robust speech recognition are of critical importance in practical deployment of speech technology. Many denoising techniques exist in the literature, e.g., [1, 2, 3]. However, few of them are based on a general and rigorous framework, while at the same time demonstrating effectiveness in large-scale robust speech recognition experiments.

In this paper, we present a new method for speech denoising and for robust speech recognition in realistic environments. Using the framework of probabilistic models allows us to integrate detailed speech models and models of realistic non-stationary noise signals in a principled manner. The framework transforms the denoising problem into a problem of Bayes-optimal signal estimation, producing minimum mean square error (MMSE) estimators of desired features of clean speech from noisy data. We describe a fast and efficient implementation of an algorithm that computes these estimators. The performance of this method is demonstrated in large scale speech recognition experiments, where it achieves significant improvements over standard methods in quasi-stationary and non-stationary noise conditions.

Notation. Throughout the paper we consider fixed, N-sample frames, and denote the time sample within a frame by a subscript $n = 0, ..., N-1$. Time domain signals are denoted by small letters, e.g., x_n. Omitting the subscript, we denote collectively $x = (x_0, ..., x_{N-1})$. The corresponding frequency domain signals are denoted by capital letters, e.g., X_k, $k = 0, ..., N-1$. The two are related by the discrete Fourier transform (DFT), $X_k = \sum_n e^{-i\omega_k n} x_n$. Omitting the subscript k, X denotes the complex N-dimensional vector

$$X = (X_0, ..., X_{N-1}) \ . \tag{1}$$

Matrices are denoted by capital bold faced letters, e.g., $\mathbf{\Sigma}$. The Gaussian distribution over X with mean $\hat{X}$ and covariance $\mathbf{\Sigma}$ is

$$\begin{aligned} p(X) &= \mathcal{N}(X \mid \hat{X}, \mathbf{\Sigma}) \\ &= \mid 2\pi\mathbf{\Sigma} \mid^{-1/2} \exp\left[-\frac{1}{2}(X-\hat{X})^\dagger \mathbf{\Sigma}^{-1}(X-\hat{X})\right] \ , \end{aligned} \tag{2}$$

where $X^\dagger = (X^\star)^T$ (complex transposition), and the covariance matrix is Hermitian ($\mathbf{\Sigma}^\dagger = \mathbf{\Sigma}$) and positive definite.

2. Bayesian Denoising

We consider the case where a single speech source is present and a single microphone is available. We use a window of length N, and assume that each N-sample frame of any signal we consider has been convolved with that window.

Let x_n be the windowed clean speech signal emitted at time n, and let y_n be the windowed noisy speech signal received at the microphone at the same time. Let u_n denote the windowed noise signal. Assuming additive noise, we have

$$y_n = x_n + u_n \ . \tag{3}$$

The basic denoising task is to provide, for each frame, an estimate $\hat{x}_n$ of the clean speech signal in that frame. In different applications, however, estimates of specific *functions* of the speech signal $f(x_0, ..., x_{N-1})$ may be desired. In particular, in the present paper we will be estimating the spectrum $S_k = \mid X_k \mid^2$ of the clean speech. Due to the variance of the estimated signal, the estimate of a function f differs from applying that function to the estimated speech signal. Thus, we define the denoising task as estimating desired functions of the clean speech.

Here is a high level view of the probabilistic modeling approach to speech denoising. In the frequency domain, (3) becomes $Y_k = X_k + U_k$. Denote the frame signals collectively by X, Y, U as in (1). We construct a probabilistic model $p_X(X)$ for the speech signal. The purpose of this model is, roughly, to quantify how likely an arbitrary signal X is to be a speech signal. We also construct a separate probabilistic model $p_U(U)$ for the noise signal. Both models are parametric and are described in detail below. Next, we need the probability distribution for observing a noisy signal Y, given that the clean signal was X. It follows from (3) that this distribution, denoted $p(Y \mid X)$, is obtained from the noise model $p_U(U)$ by substituting $U = Y - X$.

We thus have a model for the joint distribution of clean and noisy speech X, Y,

$$p(X, Y) = p_U(Y - X)p_X(X) \ . \tag{4}$$

The subscripts will henceforth be dropped and we will write $p(X)$ instead of $p_X(X)$, etc., as long as there is no ambiguity.

At the focus of probabilistic denoising is the probability distribution for the clean speech signal being X, given that the noisy signal Y has been observed. This distribution, denoted $p(X \mid Y)$ and termed the *posterior distribution* over X, is computed from (4) by Bayes' rule, $p(X \mid Y) = p(X, Y)/p(Y)$. Having obtained the posterior, we can use it to estimate the clean speech signal or a function $f(X)$ thereof by computing its average w.r.t. this posterior,

$$\hat{f} = \int f(X)p(X \mid Y)dX \ . \tag{5}$$

It can shown that the estimate of f given by $\hat{f}$ is optimal in the MMSE sense. Of course, other optimality criteria may in principle be used. For example, a natural criterion in the Bayesian framework is the likelihood of f given the data, $p(f \mid Y)$. However, computing this likelihood and maximizing it w.r.t. f is generally a difficult problem. In this paper, we use the estimator (5).

3. Speech and Noise Models

A key point of this paper is the use of a strong speech model. Via this model, the denoising algorithm incorporates prior knowledge about the structure of speech signals, which is essential to its performance. Signal processing methods such as spectral subtraction do not employ any model of speech. Much of the past work on speech denoising from a probabilistic perspective has employed very simple models, most commonly based on AR or ARMA descriptions [1]. These models are weak in the sense that they include little information on the properties of speech. Such an approach may allow features of the frame signals X (e.g., their spectra) to have arbitrary values, including values that are unlikely to occur in a speech signal. A strong model, in contrast, would weigh different values according to their likelihood. The lack of prior information could be especially significant in the single microphone case, where N clean samples need to be estimated from N noisy samples, a problem which is just barely constrained.

The most detailed statistical speech models currently available are those employed by state-of-the-art speech recognition engines. These systems are generally based on mixture of diagonal Gaussian models in the mel-cepstral domain, endowed with temporal Markov dynamics, and have a very large (~ 100000) number of states corresponding to individual atoms of speech. However, in the mel-cepstral domain the noisy speech has a strong non-linear relationship to the clean speech, making the denoising problem harder.

In this paper, we work in the frequency domain where the clean and noisy speech are related linearly. We employ a probabilistic speech model, and take an intermediate approach regarding the model structure and its size.

Mixture model for speech. Speech signals are non-stationary, meaning that the signal X at different frames may have different statistical properties. This feature can be captured using a mixture model. The model has S components, labeled $s = 1, ..., S$. Component s is a Gaussian distribution with mean zero and covariance matrix $\mathbf{A}_s$. The prior probability of component s is π_s, normalized such that $\sum_s \pi_s = 1$.

This description constitutes a generative model for speech. To generate the speech signal for a given frame, (1) select a component s with probability $p(s)$, (2) sample a N-dimensional vector X from the Gaussian distribution $p(X \mid s)$, where

$$p(X \mid s) = \mathcal{N}(X \mid 0, \mathbf{A}_s) \ , \quad p(s) = \pi_s \ . \tag{6}$$

Eq. (6) defines a mixture model for the speech distribution $p(X) = \sum_s p(X \mid s)p(s)$. Notice that the distribution (6) allows a general correlation structure between the frequency components X_k of the speech signal.

Mixture model for noise. Like speech signals, realistic noise signals are characterized by non-Gaussianity and non-stationarity. The noise model used in this paper has therefore a similar structure to the speech model. It is a mixture model with C components. Each component $c = 1, ..., C$ is a zero-mean Gaussian with a covariance matrix $\mathbf{B}_c$, i.e., $p(U \mid c) = \mathcal{N}(U \mid 0, \mathbf{B}_c)$, and a prior probability η_c.

To derive a denoising algorithm, we must determine $p(Y \mid X)$. This conditional is given by summing over the noise model components, $p(Y \mid X) = \sum_c p(Y \mid X, c)p(c)$, where

$$p(Y \mid X, c) = \mathcal{N}(Y \mid X, \mathbf{B}_c) \ , \quad p(c) = \eta_c \ . \tag{7}$$

Finally, the joint distribution of all model variables is given by

$$p(Y, X, c, s) = p(Y \mid X, c)p(X \mid s)p(c)p(s) \ . \tag{8}$$

4. Denoising Algorithm

The focus of our denoising algorithm is the clean speech posterior, given in our model by summing over all possible configurations (c, s) of speech and noise components,

$$p(X \mid Y) = \sum_{cs} p(X \mid c, s, Y)p(c, s \mid Y) \ . \tag{9}$$

It can be shown that the speech posterior conditioned on the configuration is a Gaussian, whose mean is linear in the data,

$$p(X \mid c, s, Y) = \mathcal{N}(X \mid \mathbf{W}_{cs}Y, \mathbf{D}_{cs}) \ , \tag{10}$$

where

$$\begin{aligned} \mathbf{W}_{cs} &= \mathbf{D}_{cs}\mathbf{B}_c^{-1} \ , \\ \mathbf{D}_{cs} &= \left(\mathbf{A}_s^{-1} + \mathbf{B}_c^{-1}\right)^{-1} \end{aligned} \tag{11}$$

The posterior probability of the configuration (c, s), denoted $\gamma_{cs} = p(c, s \mid Y)$, is given by

$$\gamma_{cs} = \frac{1}{z}\mathcal{N}(Y \mid 0, \mathbf{D}_{cs})\pi_s\eta_c \ , \tag{12}$$

where z is determined such that the normalization condition $\sum_{cs} \gamma_{cs} = 1$ is satisfied.

We can now estimate the speech signal X_k and its spectrum $S_k = \mid X_k \mid^2$ for each frame. Denote the mean speech signal conditioned on the configuration (c, s) and the data by

$$\hat{X}_{cs} = \mathbf{W}_{cs}Y \ . \tag{13}$$

Using (5), we get

$$\begin{aligned} \hat{X}_k &= \sum_{cs} \gamma_{cs}\hat{X}_k^{cs} \ , \\ \hat{S}_k &= \sum_{cs} \gamma_{cs}\left(\mid \hat{X}_k^{cs} \mid^2 + (\mathbf{D}_{cs})_{kk}\right) \ . \end{aligned} \tag{14}$$

Hence, the denoised speech signal $\hat{X}_k$ is obtained by linearly transforming the data, where the total transformation $\sum_{cs} \gamma_{cs} \mathbf{W}_{cs}$ combines additively the transformation corresponding to the different configurations, weighted by the configuration posterior. This posterior changes from frame to frame, making the filter time-varying. The denoised spectrum $\hat{S}_k$ is obtained in a similar manner. Notice that it differs from $\mid \hat{X}_k \mid^2$ due to the variance in the posterior $p(c, s, X \mid Y)$.

5. Efficient Implementation

Whereas the speech model (6) can describe quite general distributions, learning general $N \times N$ covariance matrices $\mathbf{A}_s$ from data may be a complex task. In this paper, we use a parametrization based on the linear prediction coding (LPC) model of speech production. Mathematically, this is an autoregressive model of order p, which describes the windowed time domain signal x_n. In this model, the signal at time point n is given by a linear combination of the signals at the preceding p time points, plus a noise signal. We assume a different LPC model for each speech cluster s, given by

$$x_n = \sum_{m=1}^{p} \theta_m^s x_{n-m} + v_n \ . \tag{15}$$

The coefficients θ_m^s are related to the physical shape of the vocal tract. v_n is termed *excitation noise*. It has mean zero, and its variance in cluster s is denoted by ν_s. We will describe the covariance matrix $\mathbf{A}_s$ in terms of the LPC parameters $(\theta_1^s, ..., \theta_p^s, \nu_s)$. This approach is related to that of [2].

We turn the description (15) into a probabilistic model in the frequency domain in two stages. First, we assume that the excitation noise is a temporally independent Gaussian, which leads to a Gaussian distribution for the time domain signal of the form

$$p(x \mid s) = \prod_{n=0}^{N-1} \mathcal{N}(x_n \mid \sum_{m=1}^{p} \theta_m^s x_{n-m}, \nu_s) \ . \tag{16}$$

Next, we switch to the frequency domain by applying DFT in (16) using the identity

$$\sum_{n=0}^{N-1} \nu_s^{-1} (x_n - \sum_{m=1}^{p} \theta_m^s x_{n-m})^2 = \sum_{k=0}^{N-1} (A_k^s)^{-1} \mid X_k \mid^2 \ , \tag{17}$$

where A_k^s is defined by

$$A_k^s = N\nu_s / \mid \Theta'_{s,k} \mid^2 \ , \tag{18}$$

and $\Theta'_{s,k}$ is the N-point DFT of $(1, -\theta_1^s, ..., -\theta_p^s)$.

Hence, the distribution of the frequency domain signal in cluster s is given by $p(X \mid s) = \mathcal{N}(X \mid 0, \mathbf{A}_s)$ as in (6), but with a diagonal covariance matrix

$$(\mathbf{A}_s)_{kl} = A_k^s \delta_{kl} \ . \tag{19}$$

It is easy to show that the diagonal elements A_k^s form the mean spectrum of the signals in cluster s, i.e., $\langle \mid X_k \mid^2 \rangle = A_k^s$, where the average is taken w.r.t. $p(X \mid s)$.

We point out that the frequency components X_k are now mutually independent, given the cluster label s. Moreover, the real and imaginary parts of X_k are also independent and have the same variance. This follows from the LPC structure of our model, and reflects the fact that the model describes the speech spectrum but not its phase.

In a similar fashion, we describe the covariance matrices $\mathbf{B}_c$ of the noise model (7) in terms of the LPC parameters $(\phi_1^s, ..., \phi_q^s, \lambda_s)$. This results in diagonal matrices, where the noise spectrum of component c, $B_k^c = N\lambda_c / \mid \Phi'_{c,k} \mid^2$, is on the diagonal.

Observe that the linear transformation $\mathbf{W}_{cs}$ in (10,11), which is now diagonal, is simply the Wiener filter corresponding to the signal and noise spectra A_k^s and B_k^c, respectively. Hence, the computation of the estimators $\hat{X}_k$ and $\hat{S}_k$ for each frame can be performed efficiently and very fast.

6. Speech Model Training

We trained the speech model parameters $\{(\theta_1^s, ..., \theta_p^s, \nu_s), s = 1, ..., S\}$ using 10000 sentences of the Wall Street Journal corpus, recorded with a close-talking microphone for 150 male and female speakers of North American English. We used 410-point frames with a 160-point overlap at a sampling rate of 16kHz, and employed a $N = 512$-point DFT after applying a Hamming window. The LPC model in each component had the order $p = 14$, and $S = 256$ components were used.

Before describing our training procedure we define some notation. Let T denote the number of speech frames in our dataset, and let X^t be the frequency domain speech signal in frame t, where $t = 1, ..., T$. Let ρ_s^t denote the posterior probabilities $\rho_s^t = p(s \mid X^t)$ of component labels s at frame t. Notice that they differ from the posterior probabilities γ_{cs} in (12), which are computed from the noisy data. In addition, let Q_k^s be the speech power spectrum corresponding to component s, defined by

$$Q_k^s = \sum_t \rho_s^t \mid X_k^t \mid^2 / \sum_t \rho_s^t \ . \tag{20}$$

We train the model using an EM algorithm which proceeds as follows. In the E-step, we compute the posterior probabilities $\rho_s^t \propto p(X^t \mid s)p(s)$, where $p(X^t \mid s)$ and $p(s)$ are those in (6), and ρ_s^t is normalized such that $\sum_s \rho_s^t = 1$.

In the M-step, we update the parameters θ_m^s by solving a Levinson-Durbin equation of order p, whose autocorrelation coefficients are obtained from Q_k^s in (20) by inverse DFT. The ν_s are then updated by $\nu_s = N^2 / \sum_k \mid \Theta'_{s,k} \mid^2 Q_k^s$, where $\Theta'_{s,k}$ is the N-point DFT of $(1, -\theta_1^s, ..., -\theta_p^s)$. Finally, the mixing fractions are updated as usual via $\pi_s = \sum_t \rho_s^t / T$.

The model parameters were initialized as follows. From each frame in the dataset, the parameters $(\theta_1, ..., \theta_p, \nu)$ were extracted by solving the Levinson-Durbin equation, where the autocorrelation was obtained from the frame spectrum $\mid X_k \mid^2$. These parameters were converted into cepstral coefficients, and clustered into S classes using K-means clustering. The resulting hard clusters induce a corresponding clustering of the speech frames X^t in the dataset. For each cluster s, we computed the parameters $(\theta_1^s, ..., \theta_p^s, \nu_s)$ as in the M-step above, using $\rho_s^t = 0, 1$ that corresponds to hard clusters. These parameters were used as initial values. The parameters π_s were initialized to the relative number of frames in each hard cluster.

7. Experimental Results

To examine the effectiveness of this algorithm in robust speech recognition, we used the denoised speech spectra $\hat{S}_k$ (14) to compute the input signals to a recognition system. Two paradigms are used. In the first one, the recognizer was trained on clean speech. In the second one, the recognizer was retrained on a training set consisting of signals that were computed from the denoised spectra.

The large vocabulary continuous recognition system used in our experiments is a version of the Microsoft continuous-density HMMs (Whisper) with 6000 tied HMM states (senones), 20 Gaussians per state, which uses a speech representation consisting of Mel-cepstrum (MFCC), delta cepstrum, and delta-delta cepstrum. A fixed, bigram language model is used in all the experiments. The system had been trained on a total of $17,809$ female clean speech sentences.

The test set consisted of 167 female WSJ sentences, which were distorted by adding synthetic white non-Gaussian noise, quasi-stationary noise recorded in an office, or non-stationary noise recorded at an airport near a plane whose engine is shutting off gradually. The amount of noise added to the clean speech sentences was determined by a pre-specified SNR. The denoising algorithm was applied to these data and produced an estimate of the clean speech spectra. The MFCCs were computed from these spectra, and the delta and delta-delta cepstra were then computed from the MFCCs. The resulting speech representation was fed to the recognizer.

Table 1 shows the results for white noise at 10 dB SNR. A single component noise model was used. On the bottom are three word error rate (WER) baselines. Preprocessing the test set by a spectral subtraction algorithm (described in [3]) gives a WER of 33.79%, and retraining on the output of spectral subtraction reduces it to 11.74%. Preprocessing the test set by the new algorithm using a $S = 64$-component speech model has a WER of 19.21%, which decreases to 12.81% for $S = 256$. Retraining on the output of this algorithm reduces the WER further to 10.34%, which is an excellent result.

Table 1: *Recognition performance comparison (WER %) for WSJ speech data corrupted by white noise at 10 dB SNR.*

Systems	WERs
new algorithm, S=64	19.21
new algorithm, S=256	12.81
new algorithm + retraining, S=256	10.34
spectral subtraction	33.79
spectral subtraction + retraining	11.74
no preprocessing (noisy-matched)	14.03
no preprocessing (mismatched)	55.06
clean speech	4.87

We also explored the effects of approximations on recognition and on computational speed (table 2). The approximation was carried out by choosing a small number of the largest terms in the weighted sum of (14), using a $S = 256$-component speech model and a single component noise model.

Table 2: *WERs (%) and computation speed for approximating the denoising algorithm.*

Number of terms	WERs	Computation speed
256	12.81	4.01 $\times$ Real time
5	13.48	3.80 $\times$ Real time
3	14.51	3.66 $\times$ Real time
1	18.54	3.45 $\times$ Real time

Table 3 shows the results for a quasi-stationary office noise at -5 dB SNR, and table 4 shows the results for a highly time-varying noise at 10 dB SNR of a plane engine shutting down. A $C = 4$-component noise model was used in the latter.

Table 3: *Recognition performance comparison (WER %) for WSJ speech data corrupted by office noise at -5 dB SNR.*

Systems	WERs
new algorithm, S=64	15.07
new algorithm + retraining, S=64	7.50
spectral subtraction	14.15
spectral subtraction + retraining	7.05
no preprocessing (noisy-matched)	7.27
no preprocessing (mismatched)	20.16
clean speech	4.87

Table 4: *Recognition performance comparison (WER %) for WSJ speech data corrupted by plane engine noise at 10 dB SNR.*

Systems	WERs
new algorithm, S=256	11.60
spectral subtraction	25.04
spectral subtraction + retraining	10.12
no preprocessing (noisy-matched)	9.71
no preprocessing (mismatched)	28.77
clean speech	4.87

8. Conclusion

We have presented a new and quite general framework for speech denoising and robust speech recognition. We demonstrated a fast and efficient implementation of this framework, and obtained very good recognition results. This implementation uses a speech model where the spectra, in effect, are smoothed using the LPC parameters. Other smoothing methods may also be used, such as methods based on cepstra or on specific filter banks. We are currently extending this framework in several directions, including modeling the effect of reverberations. We are also pursuing approximation methods for reducing the complexity of the algorithm when large noise models are used.

9. References

[1] Dembo, A. and Zeitouni, O., Maximum a posteriori estimation of time varying ARMA processes from noisy observations, IEEE Trans. Acoustics, Speech, and Signal Processing 36(4), 471–476, 1988.

[2] Ephraim, Y., A Bayesian estimation approach for speech enhancement using hidden Markov models, IEEE Trans. Signal Processing, vol. 40, 725–735, 1992.

[3] Deng, L., Acero, A., Plumpe, M., and Huang, X.D., Large vocabulary speech recognition under adverse acoustic environments, Proc. International Conference on Spoken Language Processing, vol. 3, 806–809, 2000.