ALGONQUIN: Iterating Laplace's Method to Remove Multiple Types of Acoustic Distortion for Robust Speech Recognition

Brendan J. Frey[1], *Li Deng*[2], *Alex Acero*[2], *Trausti Kristjansson*[1]

[1] Probabilistic Inference Group, University of Toronto, www.cs.toronto.edu/∼frey
[2] Speech Technology Group, Microsoft Research, www.research.microsoft.com

Abstract

One approach to robust speech recognition is to use a simple speech model to remove the distortion, before applying the speech recognizer. Previous attempts at this approach have relied on unimodal or point estimates of the noise for each utterance. In challenging acoustic environments, *e.g.*, an airport, the spectrum of the noise changes rapidly during an utterance, making a point estimate a poor representation. We show how an iterative form of Laplace's method can be used to estimate the clean speech, using a time-varying probability model of the log-spectra of the clean speech, noise and channel distortion. We use this method, called ALGONQUIN, to denoise speech features and then feed these features into a large vocabulary speech recognizer whose WER on the *clean* Wall Street Journal data is 4.9%. When 10 dB of noise consisting of an airplane engine shutting down is added to the data, the recognizer obtains a WER of 28.8%. ALGONQUIN reduces the WER to 12.6%, well below the WER of 25.0% obtained by our spectral subtraction algorithm, and close to the WER of 9.7% obtained by the slow procedure of retraining the recognizer on training data corrupted by the exact same noise. In fact, if ALGONQUIN is used to denoise the noisy training data before the recognizer is retrained, the WER is improved to 8.5%. For 10 dB of additive uniform white noise, our spectral subtraction algorithm reduces the WER from 55.1% to 33.8%. ALGONQUIN reduces the WER to 14.2%. The recognizer trained on noisy data obtains a WER of 14%, whereas the recognizer trained on noisy data denoised by ALGONQUIN obtains a WER of 9.9%.

1. Introduction

Two main approaches to robust speech recognition [1] include "recognizer domain approaches" (c.f. [2, 3, 4]), where the acoustic recognition model is modified or retrained to recognize noisy, distorted speech, and "feature domain approaches" (c.f. [5, 6]), where the features of noisy, distorted speech are first denoised and then fed into a speech recognition system whose acoustic recognition model is trained on clean speech.

One advantage of the feature domain approach over the recognizer domain approach is that the speech modeling part of the denoising model can have much lower complexity than the full acoustic recognition model. This can lead to a much faster overall system, since the denoising process uses probabilistic inference in a much smaller model. Also, since the complexity of the denoising model is much lower than the complexity of the recognizer, the denoising model can be adapted to new environments more easily, or a variety of denoising models can be stored and applied as needed.

One concern about the feature domain approach is that the residual noise left over after denoising may significantly degrade recognition performance. However, in [6], it is shown that by training the recognizer on speech that is first corrupted by a *variety* of noise types and then denoised using a particular method, excellent recognition results are obtained using the denoising method on new noise types. This is because the *residual noise* left by the denoising method has statistics that tend to be similar for different noise types. This result justifies the feature domain approach we present here.

We model the log-spectra of the clean speech, noise, and channel impulse response function using mixtures of Gaussians. The relationship between these log-spectra and the log-spectrum of the noisy speech is nonlinear, leading to a posterior distribution over the clean speech that is a mixture of non-Gaussian distributions.

We show how an iterative form of Laplace's method (using the vector Taylor series to approximate a density with a Gaussian) can be used to infer the clean speech. Our method, called ALGONQUIN, improves on previous work using Laplace's method [8] by modeling the variance of the noise and channel instead of using point estimates, by modeling the noise and channel as a mixture of different types instead of one type, by iterating Laplace's method to track the clean speech instead of applying it once at the model centers, and by accounting for the error in the nonlinear relationship between the log-spectra.

From experiments on large vocabulary recognition using the Wall Street Journal data with additive white noise, office noise, and airplane engine noise, we find that ALGONQUIN obtains significantly lower WERs than a spectral subtraction method. In fact, ALGONQUIN's WERs are close to the WERs obtained by a recognizer that is retrained by adding the test noise to the training set. When the noisy training data is denoised by ALGONQUIN before retraining the recognizer, the WERs drop significantly.

2. Model of clean speech, noise, channel, and noisy distorted speech

After describing a probability model of the class of clean speech, clean speech log-spectrum, class of noise, noise log-spectrum, class of channel, channel impulse response log-spectrum, and noisy distorted speech, we show how probabilistic inference in this model can be used to estimate the clean speech.

For clarity, we present a version of ALGONQUIN that treats frames of log-spectra independently. The extension of the version presented here to HMM models of speech, noise and channel distortion is analogous to the extension of a mixture of Gaussians to an HMM with Gaussian outputs.

Following [7, 8], we derive an approximate relationship between the log spectra of the clean speech, noise, channel and noisy speech. Assuming additive noise and linear channel distortion, in the time domain we have

$$y(t) = h(t) \star x(t) + n(t), \tag{1}$$

where "$\star$" indicates convolution.

We obtain the Fourier transform for a particular frame (25 ms spaced at 10 ms intervals) by applying a window and computing the FFT. Assuming the channel frequency response is constant across each mel-frequency filter band, we obtain the mel-frequency domain relationship,

$$Y(f) \approx H(f)X(f) + N(f). \tag{2}$$

Assuming the channel impulse response is shorter than the frame size, the energy spectrum is obtained as follows:

$$\begin{aligned}
&|Y(f)|^2 \\
&\quad = Y(f)^* Y(f) \\
&\quad \approx (H(f)X(f) + N(f))^* (H(f)X(f) + N(f)) \\
&\quad = (H(f)^* H(f))(X(f)^* X(f)) + (N(f)^* N(f)) \\
&\qquad\qquad + 2\mathrm{Re}(N(f)^* H(f)X(f)), \\
&\quad = |H(f)|^2 |X(f)|^2 + |N(f)|^2 \\
&\qquad\qquad + 2N(f)^* H(f)X(f).
\end{aligned} \tag{3}$$

If the phase of the noise and the speech are uncorrelated, the last term in the above expression is small and we can approximate the energy spectrum as follows:

$$|Y(f)|^2 \approx |H(f)|^2 |X(f)|^2 + |N(f)|^2. \tag{4}$$

Letting $\mathbf{y}$ be the vector containing the log-spectrum $\log |Y(:)|^2$, and similarly for $\mathbf{h}$, $\mathbf{x}$ and $\mathbf{n}$, we can rewrite (4) as

$$\begin{aligned}
\exp(\mathbf{y}) &\approx \exp(\mathbf{h})\exp(\mathbf{x}) + \exp(\mathbf{n}) \\
&= \exp(\mathbf{h}+\mathbf{x}) + \exp(\mathbf{n}) \\
&= \exp(\mathbf{h}+\mathbf{x}) \circ (\mathbf{1} + \exp(\mathbf{n}-\mathbf{h}-\mathbf{x})),
\end{aligned} \tag{5}$$

where the $\exp()$ function operates in an element-wise fashion on its vector argument and the "$\circ$" symbol indicates element-wise product.

Taking the logarithm, we obtain a function $\mathbf{g}()$ that is an approximate mapping of $\mathbf{h}$, $\mathbf{x}$ and $\mathbf{n}$ to $\mathbf{y}$ (see [7, 8] for more details):

$$\mathbf{y} \approx \mathbf{g}([\mathbf{x}\ \mathbf{n}\ \mathbf{h}]^\mathsf{T}) = \mathbf{h}+\mathbf{x}+\ln(\mathbf{1}+\exp(\mathbf{n}-\mathbf{h}-\mathbf{x})). \tag{6}$$

"$^\mathsf{T}$" indicates matrix transpose and $\ln()$ and $\exp()$ operate on the individual elements of their vector arguments.

Assuming the errors in the above approximation are Gaussian, the observation likelihood is

$$p(\mathbf{y}|\mathbf{x},\mathbf{n},\mathbf{h}) = \mathcal{N}(\mathbf{y}; \mathbf{g}([\mathbf{x}\ \mathbf{n}\ \mathbf{h}]^\mathsf{T}), \mathbf{\Psi}), \tag{7}$$

where $\mathbf{\Psi}$ is the covariance matrix of the errors.

Using a prior $p(\mathbf{x},\mathbf{n},\mathbf{h})$, the goal of denoising is to infer the log-spectrum of the clean speech $\mathbf{x}$, given the log-spectrum of the noisy speech $\mathbf{y}$. The minimum squared error estimate of $\mathbf{x}$ is

$$\begin{aligned}
\hat{\mathbf{x}} &= \int_{\mathbf{x}} \mathbf{x} p(\mathbf{x}|\mathbf{y}), \text{ where} \\
p(\mathbf{x}|\mathbf{y}) &\propto \int_{\mathbf{n},\mathbf{h}} p(\mathbf{y}|\mathbf{x},\mathbf{n},\mathbf{h}) p(\mathbf{x},\mathbf{n},\mathbf{h}).
\end{aligned} \tag{8}$$

This inference is made difficult by the fact that the nonlinearity $\mathbf{g}([\mathbf{x}\ \mathbf{n}\ \mathbf{h}]^\mathsf{T})$ in (6) makes the posterior non-Gaussian even if the prior is Gaussian.

We model the prior $p(\mathbf{x},\mathbf{n},\mathbf{h})$ using a mixture of Gaussians. Let $s \in \{1,\ldots,N_s\}$ index the combined set of Gaussian mixture components for the speech, noise and channel distortion. We usually start with a mixture of N_x Gaussians for the speech, N_n Gaussians for the noise, and N_h Gaussians for the channel, and then combine these to produce a mixture of $N_s = N_x N_n N_h$ Gaussians on the combined vector $[\mathbf{x}\ \mathbf{n}\ \mathbf{h}]^\mathsf{T}$.

The noise model can be estimated from silence periods (the method we used in our experiments) or it can be estimated from the test utterance using a generalized expectation-maximization algorithm, where the speech model is kept fixed and the noise model is adapted to the test utterance.

We use π_s, $\boldsymbol{\mu}_s$ and $\mathbf{\Sigma}_s$ to parameterize the mixture model as follows:

$$\begin{aligned}
p(s) &= \pi_s, \\
p(\mathbf{x},\mathbf{n},\mathbf{h}|s) &= \mathcal{N}([\mathbf{x}\ \mathbf{n}\ \mathbf{h}]^\mathsf{T}; \boldsymbol{\mu}_s, \mathbf{\Sigma}_s).
\end{aligned} \tag{9}$$

If the number of log-spectrum coefficients is M, then $\mathbf{x}$, $\mathbf{n}$ and $\mathbf{h}$ are M-vectors and $\boldsymbol{\mu}_s$ is a $3M$-vector. We assume the speech, noise and channel distortion are independent, so $\mathbf{\Sigma}_s$ is a diagonal $3M \times 3M$ covariance matrix.

Combining (7) and (9), the joint distribution over the mixture index, clean speech, noise, channel distortion and noisy speech is

$$\begin{aligned}
&p(\mathbf{y},\mathbf{x},\mathbf{n},\mathbf{h},s) = \\
&\mathcal{N}(\mathbf{y}; \mathbf{g}([\mathbf{x}\ \mathbf{n}\ \mathbf{h}]^\mathsf{T}), \Psi) \mathcal{N}([\mathbf{x}\ \mathbf{n}\ \mathbf{h}]^\mathsf{T}; \boldsymbol{\mu}_s, \mathbf{\Sigma}_s)\pi_s.
\end{aligned} \tag{10}$$

3. Probabilistic inference by iterating Laplace's method

For each mixture component s, we use an iterative form of Laplace's method to approximate $p(\mathbf{x},\mathbf{n},\mathbf{h}|\mathbf{y},s)$. Laplace's method uses the vector Taylor series to approximate a density with a Gaussian. Starting at the mixture center $\boldsymbol{\mu}_s$, we compute the first and second order statistics of the posterior and use these to predict the location of the mode. We iterate this procedure until convergence or for a fixed number of iterations.

In previous work [8], a single application of Laplace's method (referred to as a vector Taylor series approximation) is made at the mean of the Gaussian speech component, and the uncertainty in the noise and channel is not accounted for. For time-varying environments, accounting for variability in the noise and channel improves performance significantly.

Let ${\boldsymbol{\eta}_s}^{(i)}$ and ${\mathbf{\Phi}_s}^{(i)}$ be the mean and covariance of the current approximation at iteration i. So, at iteration i we have

$$p(\mathbf{x},\mathbf{n},\mathbf{h}|\mathbf{y},s) \approx \mathcal{N}([\mathbf{x}\ \mathbf{n}\ \mathbf{h}]^\mathsf{T}; {\boldsymbol{\eta}_s}^{(i)}, {\mathbf{\Phi}_s}^{(i)}). \tag{11}$$

Let $\mathbf{g}' : \mathcal{R}^{3M} \rightarrow \mathcal{R}^{M} \times \mathcal{R}^{3M}$ be the derivative of $\mathbf{g}()$ with respect to its argument. Since different elements in the vectors of (6) are uncoupled, this $M \times 3M$ matrix is a concatenation of three $M \times M$ diagonal matrices.

Using $\mathbf{g}'()$ to obtain a 1st order vector Taylor series expansion, we obtain the recursions

$$\begin{aligned}
&{\mathbf{\Phi}_s}^{(i+1)} = \big({\mathbf{\Sigma}_s}^{-1} + \mathbf{g}'({\boldsymbol{\eta}_s}^{(i)})^\mathsf{T} \Psi^{-1} \mathbf{g}'({\boldsymbol{\eta}_s}^{(i)})\big)^{-1}, \quad \text{and} \\
&{\boldsymbol{\eta}_s}^{(i+1)} = {\boldsymbol{\eta}_s}^{(i)} + \big({\mathbf{\Sigma}_s}^{-1} + \mathbf{g}'({\boldsymbol{\eta}_s}^{(i)})^\mathsf{T} \Psi^{-1} \mathbf{g}'({\boldsymbol{\eta}_s}^{(i)})\big)^{-1} \\
&\cdot\big({\mathbf{\Sigma}_s}^{-1}(\boldsymbol{\mu}_s - {\boldsymbol{\eta}_s}^{(i)}) + \mathbf{g}'({\boldsymbol{\eta}_s}^{(i)}) \Psi^{-1} (\mathbf{y} - \mathbf{g}({\boldsymbol{\eta}_s}^{(i)}))\big).
\end{aligned} \tag{12}$$

Initially, we set $\boldsymbol{\eta}_s{}^{(0)} = \boldsymbol{\mu}_s$ and $\boldsymbol{\Phi}_s{}^{(0)} = \boldsymbol{\Sigma}_s$.

After I iterations for every mixture component s, we use $\boldsymbol{\eta}_s{}^{(I)}$ and $\boldsymbol{\Phi}_s{}^{(I)}$ to compute the posterior responsibilities of the component indexed by s:

$$\begin{aligned}\rho_s^{(I)} = \lambda \exp\Big(&\ln \pi_s - \frac{1}{2}\ln|2\pi\boldsymbol{\Sigma}_s| + \frac{1}{2}\ln|2\pi\Phi_s^{(I)}| \\ &-\frac{1}{2}(\mathbf{y}-\mathbf{g}(\boldsymbol{\eta}_s{}^{(I)}))^{\top}\Psi^{-1}(\mathbf{y}-\mathbf{g}(\boldsymbol{\eta}_s{}^{(I)})) \\ -\frac{1}{2}tr(\boldsymbol{\Sigma}_s{}^{-1}\boldsymbol{\Phi}_s{}^{(I)}) &- \frac{1}{2}(\boldsymbol{\eta}_s{}^{(I)}-\boldsymbol{\mu}_s)^{\top}\boldsymbol{\Sigma}_s{}^{-1}(\boldsymbol{\eta}_s{}^{(I)}-\boldsymbol{\mu}_s) \\ &-\frac{1}{2}tr(\mathbf{g}'(\boldsymbol{\eta}_s{}^{(I)})^{\top}\Psi^{-1}\mathbf{g}'(\boldsymbol{\eta}_s{}^{(I)})\boldsymbol{\Phi}_s{}^{(I)})\Big). \qquad (13)\end{aligned}$$

λ is the normalizing constant that satisfies $\sum_s \rho_s^{(I)} = 1$.

The minimum squared error estimate of the clean speech, $\hat{x}$, is

$$\hat{x} = \sum_s \rho_s^{(I)}\boldsymbol{\eta}_s{}^{(I)}. \qquad (14)$$

We apply this algorithm on a frame-by-frame basis, until all frames in the test utterance have been denoised.

4. Speed

Our research C code denoises 40 frames/second, for 256 speech components, 1 noise component, no channel model, and 5 iterations of Laplace's method.

We used matrix notation in the above description mainly for brevity. Since elements of $\mathbf{x}$, $\mathbf{n}$, $\mathbf{h}$ and $\mathbf{y}$ that are in different rows *do not* interact in (6), the matrices in the above description are diagonal or block diagonal. Consequently, the operations are essentially scalar operations.

MATLAB code for this version of ALGONQUIN is available at http://www.cs.toronto.edu/∼frey/aal. The version number is ALGONQUINS2. For 256 speech components, 1 noise component, no channel model, and 5 iterations of Laplace's method, this routine is able to denoise each test utterance of the Wall Street Journal in about 2 minutes on a 1GHz Pentium machine. Several obvious tricks can be used to gain another order of magnitude in speed.

The amount of time needed by this version of ALGONQUIN scales as $\mathcal{O}(N_x \cdot N_n \cdot N_h)$, where N_x is the number of speech components, N_n is the number of noise components, and N_h is the number of channel components. We have derived a factorized variational technique that reduces the time to $\mathcal{O}(N_x + N_n + N_h)$.

5. Experimental results on large vocabulary speech recognition

We present results for noise added by computer to the Wall Street Journal data, including uniform white noise, office noise and highly time-varying noise consisting of an airplane engine shutting down. In all cases, ALGONQUIN uses a speech model consisting of a mixture of 256 Gaussians trained on the clean Wall Street Journal data.

The denoised log-spectral coefficients are converted to cepstral coefficients and then fed into the Whisper speech recognition system, which includes a language model. This system obtains a WER of 4.9% on *clean* Wall Street Journal test data.

For each type of noise, we compare the WER obtained by denoising using ALGONQUIN with the WER obtained without denoising and the WER obtained by denoising using spectral subtraction [6].

Table 1: WER on Wall Street Journal test data with uniform white noise, SNR = 10 dB.

Method	WER
No denoising	55.1%
Spectral subtraction	33.8%
ALGONQUIN	15.3%
Recognizer trained on noisy speech	14.0%
Recognizer trained on ALGONQUIN output	9.9%
Noise-free data	4.9%

Table 2: WER on Wall Street Journal test data with office noise, SNR = -5 dB.

Method	WER
No denoising	20.2%
Spectral subtraction	14.2%
ALGONQUIN	9.1%
Recognizer trained on noisy speech	7.3%
Recognizer trained on ALGONQUIN output	7.2%
Noise-free data	4.9%

The spectral subtraction technique [6] obtains a point estimate of the noise spectrum during "silence" periods and subtracts this spectrum from the noisy speech spectrum. The subtraction is thresholded to prevent the energy from going to low.

In [6], it was shown that by training the recognizer on speech that is first corrupted by a *variety* of noise types and then denoised using a particular method, excellent recognition results are obtained using the denoising method on new noise types. We are currently obtaining results for ALGONQUIN applied in this way. For now, we give the WER obtained by denoising test data with ALGONQUIN and then feeding it into a recognizer that is trained on data that had the *same* noise added and was then denoised using ALGONQUIN before training. We compare this WER with the WER obtained by training the recognizer on the noisy data without first denoising the data. In this way, we can see whether or not ALGONQUIN makes the recognition task easier for the retrained recognizer.

5.1. Uniform white noise

We added 10dB of uniform white noise to the test data. The noise model is a single Gaussian with mean and variance estimated from the first 50 silence frames of each test utterance. The results are shown in Table 1. ALGONQUIN gives a WER that is significantly lower than the WER obtained by spectral subtraction and almost as good as the WER obtained by the recognizer that is trained on noisy training data. A significant improvement in WER is obtained if the recognizer is trained on noisy data that is first denoised using ALGONQUIN.

5.2. Office noise

This noise sequence was recorded in an office and contains typical office sounds, such as air conditioning, computer fan and disk, keyboard typing and the room acoustics. The noise level is -5 dB. The noise model is a single Gaussian with mean and variance estimated from the first 50 silence frames of each test utterance. The results are shown in Table 2.

5.3. Airplane engine noise

This noise sequence was recorded at an airport and contains the highly time-varying sound of an aircraft engine shutting down,

Table 3: WER on Wall Street Journal test data with airplane engine noise, SNR = 10 dB.

Method	WER
No denoising	28.8%
Spectral subtraction	25.0%
ALGONQUIN, 1 noise component	29.4%
ALGONQUIN, 2 noise components	22.9%
ALGONQUIN, 4 noise components	18.2%
ALGONQUIN, 8 noise components	13.0%
ALGONQUIN, 16 noise components	12.6%
Recognizer trained on noisy speech	9.7%
Recognizer trained on ALGONQUIN output (8 noise components)	8.5%
Noise-free data	4.9%

cycling through harmonics that are similar to speech harmonics. We set the noise level to 10 dB. We give results for a noise model consisting of 1, 2, 4, 8 and 16 mixture components. The results are shown in Table 3. For this highly time-varying noise, the advantage of the mixture model over a point estimate or a single Gaussian is clear.

6. Discussion

ALGONQUIN is a fast technique for denoising speech spectrum features. In this paper, we used the log-spectrum features, although the cepstrum features or even the energy spectrum features can be used.

On the Wall Street Journal data set, ALGONQUIN obtains WERs that are significantly lower than a standard spectral subtraction technique [6], and comparable to the performance obtained by the cumbersome process of training the recognizer on noisy training data. The WERs obtained by the recognizer trained on noisy data *improve* if ALGONQUIN is used to denoise both the training and test data. For highly time-varying noise, we find that using a mixture model for the noisy leads to a significant improvement over a single Gaussian.

In contrast to previous work using Laplace's method [8], ALGONQUIN

- accounts for variability in the features for noise and channel distortion (instead of using a point estimate)
- accounts for the covariance between the speech features and the noise and channel features
- uses an iterative form of Laplace's method, that tracks the interpolated estimates of the speech, noise and channel (instead of being applied at the fixed model centers)
- accounts for the error in the nonlinear relationship between the noisy speech and the clean speech, noise and channel

In contrast to the recognizer domain approach of Gales and Young [4], ALGONQUIN uses a small speech model (in our experiments, 256 states) to account for the noise and channel. This means the model can be adapted more quickly to new types of noise, since the recognizer typically has about 120,000 mixture states. In contrast to Attias *et. al.*, who perform inference in a time-domain model of the speech (although computations are performed in the frequency domain for speed), ALGONQUIN performs inference in a model of the log-spectrum features.

The research C code for ALGONQUIN denoises 40 frames/second, for 256 speech components, 1 noise component, no channel model, and 5 iterations of Laplace's method. Straightforward implementation tricks can be used to reduce this time by an order of magnitude. When the number of noise components is increased, the time needed to denoise the utterance increases. We have derived a factorized variational method that will allow ALGONQUIN to run significantly faster and we are currently running experiments to verify that the variational technique gives comparable WERs.

We are also running experiments on a version of ALGONQUIN that can estimate the noise model from the test utterance, even during periods of non-"silence". The procedure of iterating Laplace's method can be viewed as maximizing an approximate lower bound on the log-probability of the test utterance. So, it can be used as an E-step in a generalized EM algorithm [10]. In the M-step, the parameters of the noise model are adjusted to increase the approximate bound on the log-probability of the test utterance.

7. References

[1] J. C. Junqua and J. P. Haton, *Robustness in Automatic Speech Recognition*, Kluwer Academic Publishers, Norwell MA., 1996.

[2] A. P. Varga and R. K. Moore, "Hidden Markov model decomposition of speech and noise," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*. 1990, pp. 845–848, IEEE Press.

[3] M. Gales, *Model-Based Techniques for Noise Robust Speech Recognition*, Cambridge University, Cambridge England, 1995, Doctoral dissertation.

[4] M. J. F. Gales and S. J. Young, "Robust continuous speech recognition using parallel model combination," *IEEE Speech and Audio Processing*, vol. 4, no. 5, pp. 352–359, 1996.

[5] S. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 27, pp. 114–120, 1979.

[6] L. Deng, A. Acero, M. Plumpe, and X. D. Huang, "Large-vocabulary speech recognition under adverse acoustic environments," in *Proceedings of the International Conference on Spoken Language Processing*, October 2000, pp. 806–809.

[7] A. Acero, L. Deng, T. Kristjansson, and J. Zhang, "HMM adaptation using vector Taylor series for noisy speech recognition," in *Proceedings of the International Conference on Spoken Language Processing*, October 2000, pp. 869–872.

[8] P. Moreno, *Speech Recognition in Noisy Environments*, Carnegie Mellon University, Pittsburgh PA, 1996, Doctoral dissertation.

[9] H. Attias, J. C. Platt, A. Acero, and L. Deng, "Speech denoising and dereverberation using probabilistic models," in *Advances in Neural Information Processing Systems 13*. MIT Press, Cambridge MA., 2001.

[10] R. M. Neal and G. E. Hinton, "A view of the EM algorithm that justifies incremental, sparse, and other variants," in *Learning in Graphical Models*, M. I. Jordan, Ed., pp. 355–368. Kluwer Academic Publishers, Norwell MA., 1998.