Speech Enhancement, Gain, and Noise Spectrum Adaptation Using Approximate Bayesian Estimation

Jiucang Hao, Hagai Attias, Srikantan Nagarajan, Te-Won Lee, *Member, IEEE*, and Terrence J. Sejnowski, *Fellow, IEEE*

***Abstract*—This paper presents a new approximate Bayesian estimator for enhancing a noisy speech signal. The speech model is assumed to be a Gaussian mixture model (GMM) in the log-spectral domain. This is in contrast to most current models in frequency domain. Exact signal estimation is a computationally intractable problem. We derive three approximations to enhance the efficiency of signal estimation. The Gaussian approximation transforms the log-spectral domain GMM into the frequency domain using minimal Kullback–Leiber (KL)-divergency criterion. The frequency domain Laplace method computes the maximum *a posteriori* (MAP) estimator for the spectral amplitude. Correspondingly, the log-spectral domain Laplace method computes the MAP estimator for the log-spectral amplitude. Further, the gain and noise spectrum adaptation are implemented using the expectation–maximization (EM) algorithm within the GMM under Gaussian approximation. The proposed algorithms are evaluated by applying them to enhance the speeches corrupted by the speech-shaped noise (SSN). The experimental results demonstrate that the proposed algorithms offer improved signal-to-noise ratio, lower word recognition error rate, and less spectral distortion.**

***Index Terms*—Approximate Bayesian estimation, Gaussian mixture model (GMM), speech enhancement.**

I. Introduction

In real-world environments, speech signals are usually corrupted by adverse noise, such as competing speakers, background noise, or car noise, and also they are subject to distortion caused by communication channels; examples are room reverberation, low-quality microphones, etc. Other than specialized studios or laboratories when audio signal is recorded, noise is recorded as well. In some circumstances such as cars in traffic, noise levels could exceed speech signals. Speech enhancement improves the signal quality by suppression of noise and reduction of distortion. Speech enhancement has many applications; for example, mobile communications, robust speech recognition, low-quality audio devices, and hearing aids.

Because of its broad application range, speech enhancement has attracted intensive research for many years. The difficulty arises from the fact that precise models for both speech signal and noise are unknown [1], thus speech enhancement problem remains unsolved [2]. A vast variety of models and speech enhancement algorithms are developed which can be broadly classified into two categories: single-microphone class and multi-microphone class. While the second class can be potentially better because of having multiple inputs from microphones, it also involves complicated joint modeling of microphones such as beamforming [2]–[4]. Algorithms based on a single microphone have been a major research focus, and a popular subclass is spectral domain algorithms.

It is believed that when measuring the speech quality, the spectral magnitude is more important than its phase. Boll proposed the spectral subtraction method [5], where the signal spectra are estimated by subtracting the noise from a noisy signal spectra. When the noisy signal spectra fall below the noise level, the method produces negative values which need to be suppressed to zero or replaced by a small value. Alternatively, signal subspace methods [6] aim to find a desired signal subspace, which is disjoint with the noise subspace. Thus, the components that lie in the complementary noise subspace can be removed. A more general task is source separation. Ideally, if there exists a domain where the subspaces of different signal sources are disjoint, then perfect signal separation can be achieved by projecting the source signal onto its subspace [7]. This method can also be applied to the single-channel source separation problem where the target speaker is considered as signal and the competing speaker is considered as noise. Other approaches include algorithms based on audio coding algorithms [8], independent component analysis (ICA) [9], and perceptual models [10].

Performance of speech enhancement is commonly evaluated using some distortion measures. Therefore, enhanced signals can be estimated by minimizing its distortion, where the expectation value is utilized, because of the stochastic property of speech signal. Thus, statistical-model-based speech enhancement systems [11] have been particularly successful. Statistical approaches require prespecified parametric models for both the signal and the noise. The model parameters are obtained by maximizing the likelihood of the training samples of the clean signals using the expectation–maximization (EM) algorithm. Because the true model for speech remains unknown [1], a variety of statistical models have been proposed. Short-time spectral amplitude (STSA) estimator [12] and log-spectral amplitude estimator (LSAE) [13] assume that the spectral co-

Manuscript received September 04, 2007; revised July 03, 2008. Current version published December 11, 2008. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Yariv Ephraim.

J. Hao is with the Institute for Neural Computation, University of California, San Diego, CA 92093-0523 USA.

H. Attias is with Golden Metallic, Inc., San Francisco, CA 94147 USA.

S. Nagarajan is with the Department of Radiology, University of California, San Francisco, CA 94143-0628 USA.

T.-W. Lee is with Qualcomm, Inc., San Diego, CA 92121 USA.

T. J. Sejnowski is with the Howard Hughes Medical Institute at the Salk Institute, La Jolla, CA 92037 USA, and also with the Division of Biological Sciences, University of California at San Diego, La Jolla, CA 92093 USA.

Digital Object Identifier 10.1109/TASL.2008.2005342

1558-7916/$25.00 © 2008 IEEE

efficients of both signal and noise obey Gaussian distribution. Their difference is that STSA minimizes the mean square error (MMSE) of the spectral amplitude while the LSAE uses the MMSE estimator of the log-spectra. LSAE is more appropriate because log-spectrum is believed more suitable for speech processing. Hidden Markov model (HMM) is also developed for clean speech. The developed HMM with gain adaptation has been applied to the speech enhancement [14] and to the recognition of clean and noisy speech [15]. In contrast to the frequency-domain models [12]–[15], the density of log-spectral amplitudes is modeled by a Gaussian mixture model (GMM) with parameters trained on the clean signals [16]–[18]. Spectrally similar signals are clustered and represented by their mixture components. Though the quality of fitting the signal distribution using the GMM depends on the number of mixture components [19], the density of the speech log-spectral amplitudes can be accurately represented with very small number of mixtures. However, this approach leads to a complex model in the frequency domain and exact signal estimation becomes intractable; therefore, approximation methods have been proposed. The MIXMAX algorithm [16] simplifies the mixing process such that the noisy signal takes the maximum of either the signal or the noise, which offers a closed-form signal estimation. Linear approximation [17], [18] expands the logarithm function locally using Taylor expansion. This leads to a linear Gaussian model where the estimation is easy, although finding the point of Taylor expansion needs iterative optimization. The spectral domain algorithms offer high quality speech enhancement while remaining low in computational complexity.

In this paper, different from the frequency-domain models [12]–[15], we start with a GMM in the log-spectral domain as proposed in [16]–[18]. Converting the GMM in the log-spectral domain into the frequency domain directly produces a mixture of log-normal distributions which causes the signal estimation difficult to compute. Approximating the logarithm function [16]–[18] is accurate only locally for a limited interval and thus may not be optimal. We propose three methods based on Bayesian estimation. The first is to substitute the log-normal distribution by an optimal Gaussian distribution in the Kullback–Leiber (KL) divergence [20] sense. This way in the frequency domain, we obtain a GMM with a closed-form signal estimation. The second approach uses the Laplace method [21], where the spectral amplitude is estimated by computing the maximum *a posteriori* (MAP). The Laplace method approximates the posterior distribution by a Gaussian derived from the second-order Taylor expansion of the log likelihood. The third approach is also based on Laplace method, but the log-spectra of signals are estimated using the MAP. The spectral amplitudes are obtained by exponentiating their log-spectra.

The statistical approaches discussed above rely on parameters estimated from the training samples that reflect the statistical properties of the signal. However, the statistics of the test signals may not match those of the training signals perfectly. For example, movement of the speakers and changes of the recording conditions are causes of mismatches. Such difficulty can be overcome by introducing parameters that adapt to the environmental changes. Gain and noise adaptation partially solves this problem [14], [15]. Different from the aspect of audio gain estimation in [12], [22] the gain here means the energy of signals corresponding to the volume of the audio. In [17], noise estimation is proposed, but the gain is fixed to 1. We propose an EM algorithm with efficient gain and noise estimation under the Gaussian approximation.

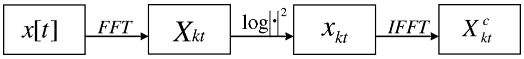


Fig. 1. Diagram for the relationship among the time domain, the frequency domain, the log-spectral domain, and the cepstral domain.

The paper is organized as the follows. In Section II, speech and noise models are introduced. In Section III, the proposed algorithms are derived in detail. In Section IV, an EM algorithm for learning gain and noise spectrum under the Gaussian approximation is presented. Section V shows the experimental results and comparisons to other methods applied to enhance the speech corrupted by speech-shaped noise (SSN). Section VI concludes the paper.

Notations: We use X or x to denote the variables derived from the clean signal, Y or y to denote the variables derived from the noisy signal, and N or n to denote the variables derived from the noise. The small letters with square brackets, $x[t]$, $y[t]$, and $n[t]$, denote time-domain variables. The capital letters, X_k, Y_k, and N_k, denote the fast Fourier transform (FFT) coefficients, the small letters, x_k, y_k, and z_k, denote the log-spectral amplitudes, and the letters with superscript c, x^c_k, y^c_k, and n^c_k, denote the cepstral coefficients. The subindex k is the frequency bin index. H denotes the gain and H^* denotes its complex conjugate. $\mathcal{N}(x \mid \mu, \nu)$ denotes the Gaussian distribution with mean μ and precision ν, which is defined as the inverse of covariance $\nu = 1/E\{(x-\mu)^2\}$. The small letter s denotes the mixture component (state index). μ_k and B_k denote the mean and the precision of the distribution for the clean signal log-spectrum x_k, and $\Gamma = \mathrm{diag}(\gamma_1, \ldots, \gamma_K)$ denotes the precision of the distribution for the noise FFT coefficients.

II. Prior Speech Model and Signal Estimation

A. Signal Representations

Let $x[t]$ be the time-domain signal. The FFT[1] coefficients X_k can be obtained by applying the FFT on the segmented and windowed signal $x[t]$. The log-spectral amplitude is computed as the logarithm of the magnitude of the FFT coefficients, $x_k = \log(|X_k|^2)$. The cepstral coefficients x^c_k are computed by taking the inverse FFT (IFFT[2]) on the log-spectral amplitudes x_k. Fig. 1 shows the relationship among different domains. Note that for the FFT coefficients, the kth component X_k is the complex conjugate of X_{K-k}. Thus, we only need to keep the first $K/2+1$ components, because the rest provides no additional information, and IFFT contains the same property. Due to this symmetry, the cepstral coefficients x^c_{kt} are real.

[1] The FFT is $X_k = \sum_{n=0}^{K-1} x[n] e^{-2\pi i k n/K}$.

[2] The IFFT is $x[n] = (1/K) \sum_{k=0}^{K-1} X_k e^{2\pi i k n/K}$.

B. Speech and Noise Models

We consider the clean signal $x[n]$ is contaminated by statistically independent and zero mean noise $n[t]$ in the time domain. Under the assumption of additive noise, the observed signal can be described by

$$y[t] = h[t] * x[t] + n[t] = \sum_m h_m x[t-m] + n[t] \quad (1)$$

where $h[t]$ is the impulse response of the filter and $*$ denotes convolution. Such signal is often processed in frequency domain by applying FFT

$$Y_k = H_k X_k + N_k \quad (2)$$

where k denotes the frequency bin and H_k is the gain. In this paper, we will focus on stationary channel, where H_k is time-independent.

Statistical models characterize the signals by its probability density function (pdf). The GMM, provided a sufficient number of mixtures, can approximate any given density function to arbitrary accuracy, when the parameters (weights, means, and covariances) are correctly chosen [19, p. 214]. The number of parameters for GMM is usually small and can be reliably estimated using the EM algorithm [19]. Here, we assume the log-spectral amplitudes $\{x_0, \ldots, x_{K-1}\}$ obey a GMM

$$p(x) = \sum_s p(x \mid s)p(s) = \sum_s \prod_k \mathcal{N}(x_k \mid \mu_{ks}, B_{ks})p(s) \quad (3)$$

where s is the state of the mixture component. For state s, $\mathcal{N}(x_k \mid \mu_{ks}, B_{ks})$ denotes a Gaussian with mean μ_{ks} and precision B_{ks} defined as the inverse of the covariance

$$\mathcal{N}(x_k \mid \mu_{ks}, B_{ks}) = \sqrt{\left|\frac{B_{ks}}{2\pi}\right|} e^{-\frac{B_{ks}}{2}(x_k - \mu_{ks})^2}. \quad (4)$$

Though each frequency bin is statistically independent for state s, they are dependent overall because the marginal density $p(x)$ does not factorize.

Use the definition of log-spectrum $x_k = \log(|X_k|^2)$, X_k can be written as $X_k = X'_k + iX''_k$, where $X'_k = e^{x_k/2}\cos\theta_k$ and $X''_k = e^{x_k/2}\sin\theta_k$ are its real part and imaginary part, θ_k is its phase. Assume that the phase is uniformly distributed $p(\theta_k) = (1/(2\pi))$, and the pdf for x_k is given in (4), we compute the pdf for the FFT coefficients as

$$\begin{aligned} p(X_k \mid s) &= p(X'_k, X''_k \mid s) \\ &= \left|\frac{\partial(X'_k, X''_k)}{\partial(x_k, \theta_k)}\right|^{-1} p(x_k \mid s)p(\theta_k) \\ &= \frac{1}{\pi|X_k|^2}\mathcal{N}(\log(|X_k|^2)|\mu_{ks}, B_{ks}) \\ &= \frac{1}{\pi|X_k|^2}\sqrt{\frac{B_{ks}}{2\pi}} e^{-\frac{B_{ks}}{2}(\log(|X_k|^2)-\mu_{ks})^2} \end{aligned} \quad (5)$$

where the Jacobian $|(\partial(X'_k, X''_k))/(\partial(x_k, \theta_k))| = e^{x_k}/2 = |X_k|^2/2$. We call this density log-normal, because the logarithm of a random variable obeys a normal distribution. The frequency-domain model is preferred compared to the log-spectral domain because of simple corruption dynamics in (2).

We consider a noise process independent on the signal and assume the FFT coefficients obey a Gaussian distribution with zero mean and precision matrix $\Gamma = \mathrm{diag}(\gamma_1, \ldots, \gamma_K)$

$$\begin{aligned} p(N) = p(Y|X) &= \prod_k \mathcal{N}(Y_k - H_k X_k|0, \gamma_k) \\ &= \prod_k \frac{\gamma_k}{\pi} e^{-\gamma_k|Y_k - H_k X_k|^2}. \end{aligned} \quad (6)$$

Note that this Gaussian density is for the complex variables. The precisions γ_k satisfy $\gamma_k = 1/E\{|Y_k - H_k X_k|^2\}$. In contrast, (4) is Gaussian density for the log-spectrum x_k which is a real random variable.

The parameters μ_{ks}, B_{ks}, and $p(s)$ of speech model given in (3) are estimated from the training samples using an EM algorithm. The details for EM algorithm can be found in [19]. The precision matrix $\Gamma = \mathrm{diag}(\gamma_1, \ldots, \gamma_K)$ of the noise model can be estimated from either pure noise or the noisy signals.

C. Signal Estimation

Under the assumption that the noise is independent on the signal, the full probabilistic model is

$$p(Y, X, s) = p(Y \mid X)p(X \mid s)p(s). \quad (7)$$

Signal estimation is done as a summation of the posterior distributions of a signal

$$p(X \mid Y) = \sum_s p(X \mid Y, s)p(s \mid Y). \quad (8)$$

For example, the MMSE estimator of a signal is given by

$$\begin{aligned} \hat{X} &= \sum_s \int X p(X \mid Y, s) dX p(s \mid Y) \\ &= \sum_s \hat{X}_s p(s \mid Y). \end{aligned} \quad (9)$$

where $\hat{X}_s$ is the signal estimator for state s. This signal estimator makes intuitive sense. Each mixture component enhances the noisy signal separately. Because the hidden state is unknown, the MMSE estimator consists of the average of the individual estimators $\hat{X}_s$, weighted by the posterior probability $p(s \mid Y)$. The block diagram is shown in Fig. 2.

The MMSE estimator suggests a general signal estimation method for the mixture models. First, an estimator based on each mixture state $\hat{X}_s$ is computed. Then the posterior state probability $p(s \mid Y)$ is calculated to reflect the contribution from state s. Finally, the system output is the summation of the estimators for the states, weighted by the posterior state probability. However, such a straightforward scheme cannot be carried out directly for the model considered. Neither the individual estimator $\hat{X}_s$ nor the posterior state probability $p(s \mid Y)$ is easy to compute. The difficulty originates from the log-normal distributions for speech in the frequency domain. We propose approximations to compute both terms. Because we assume a diagonal precision matrix for B_s in the GMM, $\hat{X}_s$ can be estimated separately for each frequency bin k.

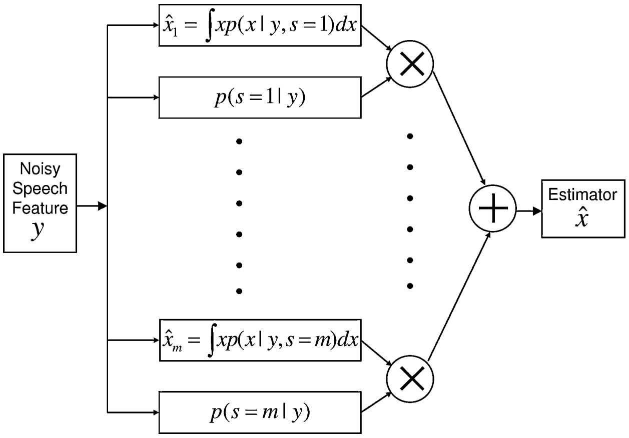

$\hat{x}_1 = \int xp(x \mid y, s=1)dx$
$p(s=1 \mid y)$
$\hat{x}_m = \int xp(x \mid y, s=m)dx$
$p(s=m \mid y)$
Noisy Speech Feature y
Estimator $\hat{x}$

Fig. 2. Block diagram for speech enhancement based on mixture models. Each mixture component enhances the signal separately. The signal estimator $\hat{x}$ is computed by the summation of individual estimator weighted by its posterior probability $p(s \mid y)$.

III. Signal Estimation Based on Approximate Bayesian Estimation

Intractability often limits the application of sophisticated models. A great amount of research has been devoted to develop accurate and efficient approximations [20], [21]. Although there are popular methods that have been applied successfully, the effectiveness of such approximations is often model dependent. As indicated in (9), two terms, $\hat{X}_s$ and $p(s \mid Y)$, are required. Three algorithms are derived to estimate both terms. One is based on Gaussian approximation. The other two methods are based on Laplace methods in the time-frequency domain and the log-spectral domain.

A. Gaussian Approximation (Gaussian)

As shown in Section II-B, the mixture of log-normal distributions for FFT coefficients makes the signal estimation difficult. If we substitute the log-normal distribution $p(X \mid s)$ in (5) by a Gaussian for each state s, the frequency domain model becomes a GMM, which is analytically tractable.

For each state s, we choose the optimal Gaussian that minimizes the KL divergence D_{KL} [23]

$$\begin{aligned} q &= \arg\min_q D_{\mathrm{KL}}(p \,\|\, q) \\ &= \arg\min_q \int p(X) \log \frac{p(X)}{q(X)}\, dX \end{aligned} \tag{10}$$

where D_{KL} is non-negative and equals to zero if and only if p equals to q almost surely. Note that D_{KL} is asymmetric about its arguments p and q, and $D_{\mathrm{KL}}(p \,\|\, q)$ is chosen because a closed-form solution for q exists.

It can be shown that the optimal Gaussian q that minimizes the KL-divergence having mean and covariance corresponding to those of the conditional probability in state s, $p(X_k \mid s)$. The mean of $p(X_k \mid s)$ is zero due the assumption of a uniform phase distribution. The second-order moments are

$$\lambda_{ks} = \int |X_k|^2 p(X_k \mid s)\, dX_k = \exp[\mu_{ks} + 1/(2B_{ks})]. \tag{11}$$

The Gaussian $q(X_k \mid s) = \mathcal{N}(X_k \mid 0, 1/\lambda_{ks})$ minimizes D_{KL}.

Under the Gaussian approximation, we have converted the GMM in log-spectral domain into a GMM in frequency domain. We denote this converted GMM by $q(X)$

$$\begin{aligned} q(X) &= \sum_s \prod_k q(X_k \mid s) p(s) \\ &= \sum_s \prod_k \mathcal{N}(X_k \mid 0, 1/\lambda_{ks}) p(s). \end{aligned} \tag{12}$$

This approach avoids the complication from the log-normal distribution and offers efficient signal enhancement.

Under the assumption of a Gaussian noise model in (6), the posterior distribution over X for state s is computed as

$$\begin{aligned} p(X_k \mid Y_k, s) &= \frac{p(Y_k \mid X_k) q(X_k \mid s)}{p(Y_k \mid s)} \\ &= \mathcal{N}(X_k \mid \hat{X}_{ks}, \phi_{ks}). \end{aligned} \tag{13}$$

It is a Gaussian with precision ϕ_{ks} and mean $\hat{X}_{ks}$ given by

$$\phi_{ks} = \lambda_{ks}^{-1} + \gamma_k \tag{14}$$

$$\hat{X}_{ks} = \frac{\gamma_k}{\phi_{ks}} Y_k \tag{15}$$

where λ_{ks} is the covariance of the speech prior and γ_k is the precision of noise pdf. Note that we have used the approximated speech prior $q(X_k \mid s)$ in (13). The individual signal estimator for each state s is given by (15).

The posterior state probability $p(s \mid Y)$ is computed

$$p(s \mid Y) = \frac{p(Y \mid s) p(s)}{p(Y)} \tag{16}$$

using the Bayes' rule. Under the speech prior $q(X \mid s)$ in (12), $p(Y \mid s)$ is computed as

$$\begin{aligned} p(Y \mid s) &= \prod_k \int p(Y_k \mid X_k) q(X_k \mid s) dX_k \\ &= \prod_k \mathcal{N}(Y_k \mid 0, \psi_{ks}) \end{aligned} \tag{17}$$

where the precision ψ_{ks} is given by

$$\psi_{ks} = \frac{1}{\lambda_{ks} + 1/\gamma_k} \tag{18}$$

Using (9) and substituting $\hat{X}_{ks}$ in (15), $p(s \mid Y)$ in (16), the signal estimation function can be written as

$$\hat{X}_k = \sum_s \hat{X}_{ks} p(s \mid Y) = \left(\sum_s \frac{\gamma_k}{\phi_{ks}} p(s \mid Y) \right) Y_k. \tag{19}$$

Each individual estimator has resembled the power response of a Wiener filter and is a linear function of Y. Note that the state probability depends on Y; therefore, the signal estimator in (19) is a nonlinear function of Y. This is analogous to a time-varying Wiener filter where the signal and noise power is known or can be estimated from a short period of the signal such as using a decision directed estimation approach [12], [22]. Here, the temporal variation is integrated through the changes of the posterior state probability $p(s \,|\, Y)$ over time.

B. Laplace Method in Frequency Domain (LaplaceFFT)

The Laplace method approximates a complicated distribution using a Gaussian around its MAP. This method suggests the MAP estimator for the original distribution which is equivalent to the more popular MMSE estimator of the resulted Gaussian. Computing the MAP can be considered as an optimization problem and many optimization tools can be applied. We use the Newton's method to find the MAP. The Laplace method is also applied to compute the posterior state probability which requires an integration over a hidden variable X. It expands the logarithm of the integrand around its mode using Taylor series expansion, and transforms the process into a Gaussian integration which has a closed-form solution. However, such a method for computing the posterior state probability is not accurate for our problem and we use an alternative approach. The final signal estimator is constructed using (9).

We derive the MAP estimator $\hat{X}_{ks}$ for each state s. The logarithm of the posterior signal pdf, conditioned on state s, is given by

$$\begin{aligned}\log p(X_k \,|\, Y_k, s) &= \log p(Y_k \,|\, X_k, s) + \log p(X_k \,|\, s) + c\\ &= -\gamma_k |Y_k - X_k|^2 + \log \frac{1}{\pi |X_k|^2}\\ &\quad - \frac{B_{ks}}{2} (\log |X_k|^2 - \mu_{ks})^2 + c \qquad (20)\end{aligned}$$

where c is a constant independent on X_k. It is more convenient to represent X_k using its magnitude r_k and phase θ_k, $X_k = r_k e^{i\theta_k}$, and we compute the MAP estimator for the magnitude r_k and phase θ_k for each state s

$$\begin{aligned}(\hat{r}_{ks}, \hat{\theta}_{ks}) &= \arg\max_{r_k, \theta_k} \{\log p(r_k, \theta_k \,|\, Y_k, s)\}\\ &= \arg\max_{r_k, \theta_k} \{\log r_k p(X_k \,|\, Y_k, s)\}. \qquad (21)\end{aligned}$$

Using (20) and neglecting the constant c, maximizing (21) is equivalent to minimizing the function h_1 defined by

$$h_1(r_k, \theta_k) = \gamma_k |Y_k - r_k e^{i\theta_k}|^2 + \frac{B_{ks}}{2} \left(\log(r_k^2) - \beta_{ks}\right)^2 \qquad (22)$$

where $\beta_{ks} = \mu_{ks} - 1/(2B_{ks})$. It is obvious from the above equation that the MAP estimator for θ_k is $\hat{\theta}_k = \angle Y_k$, which is independent on state s, and the magnitude estimator $\hat{r}_{ks}$ minimizes

$$h_1(r_k) = \gamma_k |r_{yk} - r_k|^2 + \frac{B_{ks}}{2} \left(\log(r_k^2) - \beta_{ks}\right)^2 \qquad (23)$$

where $r_{yk} = |Y_k|$. The minimization over r_k does not have an analytical solution, but it can be solved with the Newton's method. For this, we need the first-order and second-order derivatives of $h_1(r_k)$ with respect to r_k

$$h_1'(r_k) = 2\gamma_k (r_k - r_{yk}) + B_{ks} \left(\log(r_k^2) - \beta_{ks}\right) \frac{2}{r_k} \qquad (24)$$

$$h_1''(r_k) = 2\gamma_k + B_{ks} \frac{4}{r_k^2} - B_{ks} \left(\log(r_k^2) - \beta_{ks}\right) \frac{2}{r_k^2}. \qquad (25)$$

Then, the Newton's method iterates

$$\hat{r}_{ks} \leftarrow \hat{r}_{ks} - \eta \frac{h_1'(\hat{r}_{ks})}{|h_1''(\hat{r}_{ks})|}. \qquad (26)$$

The absolute value of h_1'' indicates the search of the minima of h_1. The $\eta = 1$ denotes the learning rate.

Newton's method is sensitive to the initialization and may give local minima. The two squared terms in (23) indicate that the optimal estimator $\hat{r}_{ks}$ is bounded between $e^{\beta_{ks}/2}$ and r_{yk}. We use both values to initialize $\hat{r}_{ks}$ and select the one that produces a smaller $h_1(r_k)$. Empirically, we observe that this scheme always finds a global minimum. The first term in (23) is quadratic; thus, Newton's method converges to the optimal solution faster, less than five iterations for our case, than other methods such as gradient decent.

Computing the posterior state probability $p(s \,|\, Y)$ requires the knowledge of $p(Y_k \,|\, s)$. Marginalization over X_k gives

$$p(Y_k \,|\, s) = \int p(Y_k \,|\, X_k) p(X_k \,|\, s)\, dX_k. \qquad (27)$$

However, because of the log-normal distribution $p(X_k \,|\, s)$ provided in (5), the integration cannot be solved with a closed-form answer. Either numerical methods or approximations are needed. Numerical integration is computationally expensive, leaving the approximation more efficient. We propose the following two approaches based on the Laplace method and Gaussian approximation.

1) Evaluate $p(s \,|\, Y)$ Using the Laplace Method: The Laplace method is widely used to approximate integrals with continuous variables in statistical models to facilitate probabilistic inference [21] such as computing the high order statistics. It expands the logarithm of the integrand up to its second order, leading to a Gaussian integral which has a closed-form solution. We rewrite (27) as

$$p(Y_k \,|\, s) = \int \frac{\gamma_k}{\pi} \sqrt{\frac{B_{ks}}{2\pi}} \exp(-f(X_k) - \beta_{ks})\, dX_k \qquad (28)$$

where we define

$$f(X_k) = \gamma_k |Y_k - X_k|^2 + \frac{B_{ks}}{2} (\log(|X_k|^2) - \alpha_{ks})^2 \qquad (29)$$

and $\alpha_{ks} = \mu_{ks} - 1/B_{ks}$, $\beta_{ks} = \mu_{ks} - 1/(2B_{ks})$. The Laplace method expands the logarithm of the integrand $f(X_k)$ around its minimum $\hat{X}_{ks}$ up to the second order and carries out a Gaussian integration

$$\int e^{-f(X_k)} dX_k \approx e^{-f(\hat{X}_{ks})} \sqrt{\left|\frac{2\pi}{J}\right|} \qquad (30)$$

where J is the Hessian of $f(X_k)$ evaluated at $\hat{X}_{ks}$. Denote $\hat{X}_{ks} = \hat{X}'_{ks} + i\hat{X}''_{ks}$ by its real part $\hat{X}'_{ks}$ and imaginary part $\hat{X}''_{ks}$, its magnitude by $\hat{r}_{ks} = |\hat{X}_{ks}|$. J is computed as

$$J = \begin{pmatrix} \frac{\partial^2 f}{\partial X' \partial X'} & \frac{\partial^2 f}{\partial X' \partial X''} \\ \frac{\partial^2 f}{\partial X' \partial X''} & \frac{\partial^2 f}{\partial X'' \partial X''} \end{pmatrix} \qquad (31)$$

$$= \begin{pmatrix} a_k + \frac{4\hat{X}_k'^2}{\hat{r}_{ks}^2} b_k & \frac{4\hat{X}_k' \hat{X}_k''}{\hat{r}_{ks}^2} b_k \\ \frac{4\hat{X}_k' \hat{X}_k''}{\hat{r}_{ks}^2} b_k & a_k + \frac{4\hat{X}_k''^2}{\hat{r}_{ks}^2} b_k \end{pmatrix}. \tag{32}$$

The a_k and b_k here are defined as

$$a_k = 2\gamma_k + B_{ks}\left(\log\left(\hat{r}_{ks}^2\right) - \alpha_{ks}\right)\frac{2}{\hat{r}_{ks}^2} \tag{33}$$

$$b_k = \frac{B_{ks} - B_{ks}\left(\log\left(\hat{r}_{ks}^2\right) - \alpha_{ks}\right)}{\hat{r}_{ks}^2}. \tag{34}$$

The determinant of Hessian J is

$$\det(J) = a_k^2 + 4a_k b_k. \tag{35}$$

Thus, the marginal probability is

$$p(Y_k|s) \propto \sqrt{|B_{ks}|} e^{-\beta_{ks}} e^{-f(\hat{X}_{ks})} \sqrt{\left|\frac{1}{\det(J)}\right|}. \tag{36}$$

This gives $p(s\,|\,Y)$

$$p(s\,|\,Y) = \frac{p(Y\,|\,s)p(s)}{p(Y)} \propto \prod_k p(Y_k\,|\,s)p(s). \tag{37}$$

The Laplace method in essence approximates the posterior $p(X_k\,|\,Y_k, s)$ using a Gaussian density. This is very effective in Bayesian networks, where the training set includes a large number of samples. The posterior distribution of the (hyper-) parameters has a peaky shape that closely resembles a Gaussian. The Laplace method has an error that scales as $O(T^{-1})$, where T is the number of samples [21]. However, the estimation here is based on a single sample Y. Further, the normalization factor of $p(Y_k\,|\,s)$ in (36) depends on the state s, but it is ignored. Thus, this approach does not yield good experimental results and we derive another method.

2) Evaluate $p(s\,|\,Y)$ Using Gaussian Approximation: As discussed in Section III-A, the log-normal distribution $p(X_k\,|\,s)$ has a Gaussian approximation $q(X_k\,|\,s) = \mathcal{N}(X_k\,|\,0, 1/\lambda_{ks})$ given in (12). Thus, we can compute the marginal distribution $p(Y_k\,|\,s)$ for state s as

$$\begin{aligned} p(Y_k\,|\,s) &= \int p(Y_k\,|\,X_k)p(X_k\,|\,s)dX_k \\ &\approx \int p(Y_k\,|\,X_k)q(X_k\,|\,s)\,dX_k \\ &= \mathcal{N}(0, \psi_{ks}) \end{aligned} \tag{38}$$

where the precision ψ_{ks} is given in (18). The posterior state probability $p(s\,|\,Y)$ is obtained using the Bayes' rule. It is

$$p(s\,|\,Y) = \frac{\prod_k p(Y_k\,|\,s)p(s)}{p(Y)}. \tag{39}$$

This approach uses the same procedure shown in Section III-A.

The signal estimator is the summation of the MAP estimator $\hat{r}_{ks}e^{i\angle Y_k}$ for each state s weighted by the posterior state probability $p(s\,|\,Y)$ in (39)

$$X_k = \sum_s \hat{r}_{ks} e^{i\angle Y_k} p(s\,|\,Y). \tag{40}$$

The MAP estimator for phase, $\angle Y_k$, is utilized.

C. Laplace Method in Log-Spectral Domain (LaplaceLS)

It is suggested that the human auditory system perceives a signal on the logarithmic scale, therefore log-spectral analysis such as LSAE [13] is more suitable for speech processing. Thus, we can expect better performance if the log-spectra can be directly estimated. The idea is to find the log-amplitude $\hat{v}_k = \log(|X_k|^2)$ that maximizes the log posterior probability $\log(p(X_k\,|\,Y_k, s))$ given in (20). Note that $\hat{v}_k$ is not the MAP of $p(\log(|X_k|^2)\,|\,Y_k, s)$. A similar case is LSAE [13], where the expectation of the log-spectral error is taken over $p(X)$ rather than $p(\log|X|)$. Optimization over v_k also has the advantage of avoiding negative amplitude due to local minima.

Substituting $v_k = \log(|X_k|^2)$ into (20), we compute the MAP estimator for the phase and log-amplitude v_k. Note that the optimal phase is that of the noisy signal, $\hat{\theta}_k = \angle Y_k$. The MAP estimator for the log-amplitude maximizes (20), which is equivalent to minimizing

$$h_2(v_k) = \gamma_k(r_{yk} - e^{v_k/2})^2 + v_k + \frac{B_{ks}}{2}(v_k - \mu_{ks})^2 \tag{41}$$

where $r_{yk} = |Y_k|$, and h_2 can be minimized using Newton's method. The first- and second-order derivatives are given by

$$h_2'(v_k) = -\gamma_k(r_{yk} - e^{v_k/2})e^{v_k/2} + 1 + B_{ks}(v_k - \mu_{ks}) \tag{42}$$

$$h_2''(v_k) = -\frac{1}{2}\gamma_k(r_{yk} - e^{v_k/2})e^{v_k/2} + \frac{1}{2}\gamma_k e^{v_k} + B_{ks}. \tag{43}$$

The Newton's method updates the log-amplitude v_{ks} as

$$\hat{v}_{ks} \leftarrow \hat{v}_{ks} - \eta \frac{h_2'(\hat{v}_{ks})}{|h_2''(\hat{v}_{ks})| + \tau} \tag{44}$$

where η is the learning rate, and τ is the regularization to avoid divergence when h_2'' is close to zero. This avoids the numerical instability caused by the exponential term in (41).

In the experiment, we use the noisy signal log-spectra for initialization, $\hat{v}_{ks} = \log(|Y_k|^2)$. We set $\eta = 0.5$, $\tau = 3$, and run ten Newton's iterations.

We use the same strategy as described in Section III-B.2 to compute $p(s\,|\,Y)$ using (39). The signal estimator follows

$$\bar{v}_k = \sum_s \hat{v}_{ks} p(s\,|\,Y) \tag{45}$$

$$X_k = \exp(\bar{v}_k/2)e^{i\angle Y_k}. \tag{46}$$

The MAP estimator of phase from the noisy signal is used.

In contrast to (40), where the amplitude estimators are averaged, (45) provides the log-amplitude estimator. The magnitude is obtained by taking the exponential. The exponential function is convex; thus, (45) provides a smaller magnitude estimation than (40) when $e^{\hat{v}_{ks}/2} = \hat{r}_{ks}$. Furthermore, this log-spectral estimator fits a speech recognizer, which extracts the Mel frequency cepstral coefficients (MFCCs).

IV. Learning Gain and Noise With Gaussian Approximation

One drawback of the system comes from the assumption that the statistical properties of the training set match those of the testing set, which means a lack of adaptability. However, the energy of the test signals may not be reliably estimated from a training set because of uncontrolled factors such as variations of the speech loudness or the distance between the speaker and microphone. This mismatch results in poor enhancement because the pretrained model may not capture the statistics of samples under the testing conditions. One strategy to compensate for these variations is to estimate the gain H instead of a fixed value of 1 used in the previous sections. Two conditions will be considered: frequency independent gain, which is a scalar gain and frequency dependent gain. Gain-adaptation needs to carry out efficiently. For the signal prior given in (3), it is difficult to estimate the gain because of the involvement of log-normal distributions. See Section II-B. However, under Gaussian approximation, the gain can be estimated using the EM algorithm.

Recall that the acoustic model is $Y_k = H_k X_k + N_k$ as given in (2). If $p(X_k)$ has the form of GMM and $p(N_k)$ is Gaussian, the model becomes exactly a mixture of factor analysis (MFA) model. The gain H can be estimated in the same way as estimating a loading matrix for MFA. For this purpose, we take the approach in Section III-A and approximate the log-normal pdf $p(X_k \,|\, s)$ by a normal distribution $q(X_k \,|\, s) = \mathcal{N}(X_k \,|\, 0, 1/\lambda_{ks})$, where the signal covariance λ_{ks} is given in (11). In addition, we assume additive Gaussian noise as provided in (6). Treating X_k as a hidden variable, we derive an EM algorithm, which contains an expectation step (E-step) and a maximization step (M-step), to estimate the gain H_k and the noise spectrum $\Gamma = \mathrm{diag}(\gamma_1, \ldots, \gamma_K)$.

A. EM Algorithm for Gain and Noise Spectrum Estimation

The data log-likelihood denoted by $\mathcal{L}$ is

$$\begin{aligned}\mathcal{L} &= \sum_t \log p(Y_t) = \sum_t \log\left(\sum_{s_t} \int p(Y_t, X_t, s_t) dX_t\right)\\ &\geq \sum_{ts_t} \int \tilde{q}(X_t, s_t)[\log p(Y_t, X_t, s_t) - \log \tilde{q}(X_t, s_t)]\, dX_t\end{aligned}$$

where t is the frame index. The above inequality is true for all choices of the distribution $\tilde{q}(X_t, s_t)$. When $\tilde{q}(X_t, s_t)$ equals the posterior probability $p(X_t, s_t \,|\, Y_t)$, the inequality becomes an equality. The EM algorithm is a typical technique to maximize the likelihood. It iterates between updating the auxiliary distribution $\tilde{q}(X_t, s_t)$ (E-step) and optimizing the model parameters $\{H, \Gamma\}$ (M-step), until some convergence criterion is satisfied.

The E-step computes the posterior distribution over X_t, $\tilde{q}(X_t \,|\, s_t) = p(X_t \,|\, Y_t, s_t) = \prod_k p(X_{kt} \,|\, Y_{kt}, s_t)$ with gain H fixed. And $p(X_{kt} \,|\, Y_{kt}, s_t)$ is computed as

$$p(X_{kt} \,|\, Y_{kt}, s_t) = \frac{p(Y_{kt} \,|\, X_{kt}) q(X_{kt} \,|\, s_t)}{p(Y_{kt} \,|\, s_t)}. \tag{47}$$

Note we use the approximated signal prior $q(X_{kt} \,|\, s_t)$ given in (12). Thus, the computation is a standard Bayesian inference in a Gaussian system, and one can show that $p(X_{kt} \,|\, Y_{kt}, s_t) = \mathcal{N}(X_{kt} \,|\, \tilde{X}_{kst}, \Sigma_{ks})$, whose mean $\tilde{X}_{kst}$ and precision Σ_{ks} are given by

$$\Sigma_{ks} = H_k^2 \gamma_k + 1/\lambda_{ks} \tag{48}$$

$$\tilde{X}_{kst} = \frac{\gamma_k H_k^* Y_{kt}}{\Sigma_{ks}}. \tag{49}$$

Here, H^* denotes the complex conjugate of H. We point out that the precisions are time-independent while the means are time dependent.

The posterior state probability $\tilde{q}(s_t) = p(s_t \,|\, Y_t)$ is computed as

$$\begin{aligned}\tilde{q}(s_t) = p(s_t \,|\, Y_t) &= \frac{p(Y_t \,|\, s_t) p(s_t)}{p(Y_t)},\\ &\propto \prod_k \mathcal{N}\left(Y_{kt} | 0, \frac{1}{H_k^2 \lambda_{ks} + 1/\gamma_k}\right) p(s_t).\end{aligned} \tag{50}$$

The M-step updates the gain H and noise spectrum $\Gamma = \mathrm{diag}(\gamma_1, \ldots, \gamma_K)$ with $\tilde{q}$ fixed. Now we consider two conditions: frequency-dependent gain and frequency-independent gain.

Frequency Independent Gain: H is scalar, its update rule is

$$H = \frac{\sum_{t,s_t,k} \tilde{q}(s_t) \gamma_k Y_{kt} \tilde{X}^*_{ks_t t}}{\sum_{t,s_t,k} \tilde{q}(s_t) \gamma_k \left(\tilde{X}_{ks_t t} \tilde{X}^*_{ks_t t} + \Sigma_{ks}^{-1}\right)}. \tag{51}$$

Frequency Dependent Gain: $H = \{H_1, \ldots, H_K\}$ is a vector. The update rule is, for $k = \{1, \ldots, K\}$,

$$H_k = \frac{\sum_{t,s_t} \tilde{q}(s_t) Y_{kt} \tilde{X}^*_{ks_t t}}{\sum_{t,s_t} \tilde{q}(s_t) \left(\tilde{X}_{ks_t t} \tilde{X}^*_{ks_t t} + \Sigma_{ks}^{-1}\right)}. \tag{52}$$

The update rule for the precision of noise γ_k is

$$1/\gamma_k = \frac{1}{T} \sum_{t,s_t} \int \tilde{q}(X_{kt}, s_t | Y_t) |Y_{kt} - H_k X_{kt}|^2 dX_{kt}. \tag{53}$$

The goal of the EM algorithm is to provide an estimation for the gain and the noise spectrum. Note that it is not necessary to compute the intermediate results $\tilde{X}_{ks_t t}$ in every iteration. Thus, substantial computation can be saved if we substitute (49) into the learning rules. This significantly improves the computational efficiency and saves memory. After some mathematical manipulation, the EM algorithm for the frequency dependent gain is as follows.

1) Initialize H_k and γ_k.

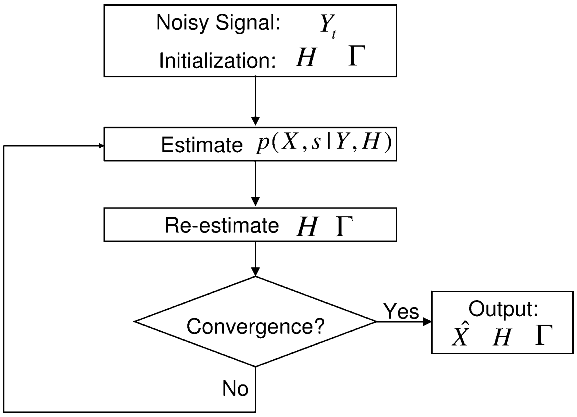


Fig. 3. Block diagram of EM algorithm for the gain and noise spectrum estimation. The E-step, computing $p(X, s \mid Y, H)$, and M-step, updating H and Γ, iterate until convergence.

2) Compute $\tilde{q}(s_t)$ using (50).
3) Update the precisions Σ_{ks} using (48).
4) Update the gain

$$H_k \leftarrow \frac{\sum_{ts_t} \tilde{q}(s_t)\Sigma_{ks}^{-1} H_k \gamma_k |Y_{kt}|^2}{\sum_{ts_t} \tilde{q}(s_t)\left(\left(\Sigma_{ks}^{-1}\gamma_k\right)^2 H_k^2 |Y_{kt}|^2 + \Sigma_{ks}^{-1}\right)}. \qquad (54)$$

5) Update the noise precision

$$\frac{1}{\gamma_k} \leftarrow \frac{1}{T}\sum_{ts_t} \tilde{q}(s_t)\left(\left(1 - \Sigma_{ks}^{-1}\gamma_k H_k^2\right)|Y_{kt}|^2 + \Sigma_{ks}^{-1} H_k^2\right). \qquad (55)$$

6) Iterate step 2), 3), 4), and 5) until convergence.

For frequency-independent gain, the gain is updated as follows:

$$H \leftarrow \frac{\sum_{ts_t k} \tilde{q}(s_t)\Sigma_{ks}^{-1} H_k \gamma_k^2 |Y_{kt}|^2}{\sum_{ts_t k} \tilde{q}(s_t)\gamma_k\left(\left(\Sigma_{ks}^{-1}\gamma_k\right)^2 H_k^2 |Y_{kt}|^2 + \Sigma_{ks}^{-1}\right)}. \qquad (56)$$

The block diagram is shown in Fig. 3. In the above EM algorithm, Σ_{ks} is time independent; thus, it is computed only once for all the frames, and $|Y_{kt}|^2$ is computed in advance.

In our experiment, because the test files are 1–2 seconds long segments, the parameters can not be reliably learned using a single segment. Thus, we concatenate four segments as a testing file. The gain is initialized to be 1. The noise covariance is initialized to be 30% of the signal covariance for all signal-to-noise ratio (SNR) conditions, which does not include any prior SNR knowledge. Because the EM algorithm for estimating the gain and noise is efficient, we set strict convergence criteria: a minimum of 100 EM iterations, the change of likelihood less than 1 and the change of gain less than 10^{-4} per iteration.

B. *Identifiability of Model Parameters*

The MFA is not identifiable because it is invariant under the proper rescaling of the parameters. However, in our case, the parameters H and Γ are identifiable, because the model for speech, a GMM trained by clean speech signals, remains fixed during the learning of parameters. The fixed speech prior removes the scaling uncertainty of the gain H. Second, the speech model is a GMM while the noise is modeled by a single Gaussian. The structure of speech, captured by the GMM through its higher order statistics, does not resemble a single Gaussian. This makes the noise spectrum Γ identifiable. As shown in our experiments, the gain H and noise spectrum Γ are reliably estimated using the EM algorithm.

V. Experiments and Results

We evaluate the performances of the proposed algorithms by applying them to enhance the speeches corrupted by various levels of SSN. The SNR, spectral distortion (SD), and word recognition error rate serve as the criteria to compare them with the other benchmark algorithms quantitatively.

A. *Task and Dataset Description*

For all the experiments in this paper, we use the materials provided by the speech separation challenge [24]. This data set contains six-word sentences from 34 speakers. The speech follows the sentence grammar, ⟨\$command⟩ ⟨\$color⟩ ⟨\$preposition⟩ ⟨\$letter⟩ ⟨\$number⟩ ⟨\$adverb⟩. There are 25 choices for the letter (*a–z except w*), ten choices for the number (*0–9*), four choices for the command (*bin, lay, place, set*), four choices for the color (*blue, green, red, white*), four choices for the preposition (*at, by, in, with*), and four choices for the adverb (*again, now, please, soon*). The time-domain signals are sampled at 25 kHz. Provided with the training samples, the task is to recover speech signals and recognize the key words (*color, letter, digit*) in the presence of different levels of SSN. Fig. 4 shows the speech and the SSN spectrum averaged over a segment under 0-dB SNR. The average spectra of the speech and the noise have the similar shape; hence, the name speech-shaped noise. The testing set includes the noisy signals under four SNR conditions, -12 dB, -6 dB, 0 dB, and 6 dB, each consisting of 600 utterances from 34 speakers.

B. *Training the Speech Model*

The training set consists of clean signal segments that are 1–2 seconds long. They are used to train our prior speech model. To obtain a reliable speech model, we randomly concatenate 2 minutes of signals from the training set and analyze them using Hanning windows, each of size 800 samples and overlapping by half of the window. Frequency coefficients are obtained by performing a 1024 points FFT to the time-domain signals. Coefficients in the log-spectral domain are obtained by taking the logarithm of the magnitude of the FFT coefficients. Due to FFT/IFFT symmetry, only the first 513 frequency components are kept. Cepstral coefficients are obtained by applying IFFT on the log-spectral amplitudes.

The speech model for each speaker is a GMM with 30 states in the log-spectral domain. First, we take the first 40 cepstral coefficients and apply a k-mean algorithm to obtain $k = 30$ clusters. Next, the outputs of the k-mean clustering are used to initialize the GMM on those 40 cepstral coefficients. Then, we convert the GMM from the cepstral domain into the log-spectral domain using FFT. Finally, the EM algorithm initialized by the converted GMM is used to train the GMM in the log-spectral domain. After training, this log-spectral domain GMM with 30 states for speech is fixed when processing the noisy signals.

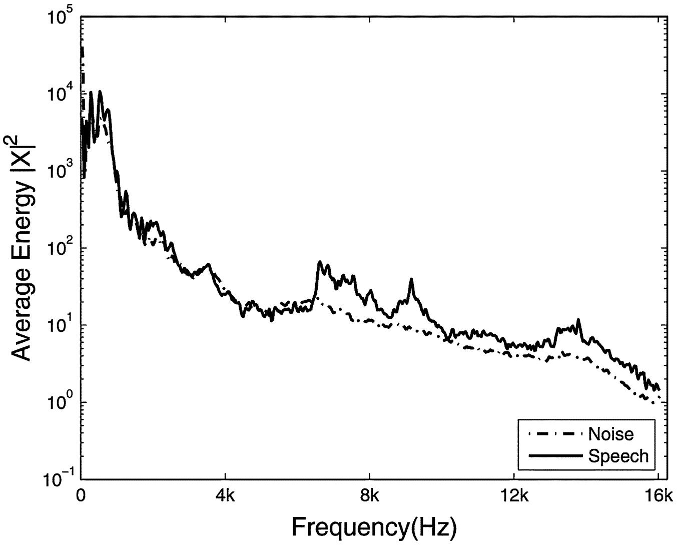


Fig. 4. Plot of SSN spectrum (dotted line) and speech spectrum (solid line) averaged over one segment under 0-dB SNR. Note the similar shapes.

C. Benchmark Algorithms for Comparison

In this section, we present the benchmark algorithms with which we compare the proposed algorithms: the Wiener filter, the perceptual model [10], the linear approximation [17], [18], and the model based on super Gaussian prior [25]. We assume that parameters of the model for noise are available, and they are estimated by concatenating 50 segments in the experiment.

1) Wiener Filter (Wiener): Time-varying Wiener filter assumes that both of the signal and noise power are known, and they are stationary for a short period of time. In the experiment, we first divide the signals into frames of 800 samples long with half overlapping. Both speech and noise are assumed to be stationary within each frame. To estimate speech and noise power, for each frame, the 200-sample-long subframes are chosen with half overlapping. On the subframes, Hanning windows are applied. Then, 256 points FFT are performed on those subframes to obtain the frequency coefficients. The power of signal within each frame t for frequency bin k, denoted by P_{tk}^x, is computed by averaging the power of FFT coefficients over all the subframes that belong to the frame t. The same method is used to compute the noise power denoted by P_{tk}^n. The signal estimation is computed as

$$X_{tjk} = \frac{P_{tk}^x}{P_{tk}^x + P_{tk}^n} Y_{tjk} \tag{57}$$

where j is the subframe index and k denotes the frequency bins. After IFFT, in the time domain, each frame can be synthesized by overlap-adding the subframes, and the estimated speech signal is obtained by overlap-adding the frames.

Because the signal and noise powers are derived locally for each frame from the speech and noise, the Wiener filter contains strong speech prior in detail. Its performance can be regarded as a sort of experimental upper bound for the proposed methods.

2) Perceptual Model (Wolfe): The perceptually motivated noise reduction technique can be seen as a masking process. The original signal is estimated by applying some suppression rules. For comparison, we use the method described in [10]. The algorithm estimates the spectral amplitude by minimizing the following cost function:

$$C(\hat{a}_k, a_k) = \begin{cases} \left(\hat{a}_k - a_k - \frac{m_k}{2}\right)^2 - \left(\frac{m_k}{2}\right)^2, & \\ & \text{if } \left|\hat{a}_k - a_k - \frac{m_k}{2}\right| > \frac{m_k}{2} \\ 0, & \text{otherwise.} \end{cases} \tag{58}$$

where $\hat{a}_k$ is the estimated spectral amplitude, and a_k is the true spectral amplitude. This cost function penalizes the positive and negative errors differently, because positive estimation errors are perceived as additive noise and negative errors are perceived as signal attenuation [10]. The stochastic property of speech is that real spectral amplitude is unavailable; therefore, $\hat{a}_k$ is computed by minimizing the expected cost function

$$\hat{a}_k = \arg\min_{\hat{a}_k} \int\int C(\hat{a}_k, a_k) p(\alpha_k, a_k \,|\, Y_k)\, d\alpha_k\, da_k \tag{59}$$

where α_k is the phase, and $p(\alpha_k, a_k \,|\, Y_k)$ is the posterior signal distribution. Details of the algorithm can be found in [10]. The MATLAB code is available online [26]. The original code adds synthetic white noise to the clean signal, we modified it to add SSN to corrupt a speech at different SNR levels.

The reason we chose this method is because we hypothesize that this spectral analysis-based approach fails to enhance the SSN corrupted speech, due to the spectral similarity between the speech and noise as shown in Fig. 4. This method, motivated from a different aspect by human perception, also serves as a benchmark with which we can compare our methods.

3) Linear Approximation (Linear): It can be shown that the relationship among the log-spectra of the signal x_k, the noisy signal y_k, and the noise n_k is given by [17], [18]

$$y_k = x_k + \log(1 + \exp(n_k - x_k)) + \epsilon_k \tag{60}$$

where ϵ_k is an error term.

The speech model remains the same which is GMM given by (3), but the noise log-spectrum n has a Gaussian density with the mean ρ and precision D, while the error term ϵ obeys a Gaussian with zero-mean and precision R

$$p(n) = \mathcal{N}(n \,|\, \rho, D) = \prod_k \mathcal{N}(n_k \,|\, \rho_k, D_k) \tag{61}$$

$$p(\epsilon) = \mathcal{N}(\epsilon \,|\, 0, R) = \prod_k \mathcal{N}(\epsilon_k \,|\, 0, R_k). \tag{62}$$

This essentially assumes a log-normal pdf for the noise FFT coefficients, in contrast to the noise model in (6).

Linear approximation to (60) has been proposed in [17] and [18] to enhance the tractability. Note that there are two hidden variables x_k and n_k due to the error term ϵ_k. Let $z_k = (x_k, n_k)^T$. Define $g(z_k) = x_k + \log(1+\exp(n_k - x_k))$ and its derivatives $g_x'(z_k) = (\partial g)/(\partial x_k) = (1/(1+\exp(n_k - x_k)))$, $g_n'(z_k) = (\partial g)/(\partial n_k) = (1/(1 + \exp(x_k - n_k)))$, $g'(z_k) = (g_x'(z_k), g_n'(z_k))^T$. Using (60) and expanding $g(z_k)$ around $\tilde{z}_{ks} = (\tilde{x}_{ks}, \tilde{n}_{ks})^T$ linearly, y_k becomes a linear function of z_k

$$y_k \approx l(z_k) + \epsilon_k \tag{63}$$

where

$$l(z_k) = g(\tilde{z}_{ks}) + g'(\tilde{z}_{ks})^T(z_k - \tilde{z}_{ks}). \tag{64}$$

The choice for $\tilde{z}_{ks}$ will be discussed later. Now we have a linear Gaussian system and the posterior distribution over z_k is Gaussian, $\mathcal{N}(z_k \mid \hat{z}_{ks}, \Lambda)$. The mean $\hat{z}_{ks}$ and the precision Λ satisfy

$$\Lambda(z_k - \hat{z}_{ks}) = -R_k(y_k - l(z_k))g'(\tilde{z}_{ks}) - G_{ks}(\zeta_{ks} - z_k) \tag{65}$$

$$\Lambda = g'(\tilde{z}_{ks})R_k g'(\tilde{z}_{ks})^T + G_{ks} \tag{66}$$

where $\zeta_{ks} = (\mu_{ks}, \rho_k)^T$ the means of GMM for the speech and noise log-spectrum, and $G_{ks} = \mathrm{diag}(B_{ks}, D_k)$ the precisions.

The accuracy of linear approximation strongly depends on the point $\tilde{z}_{ks}$ which is the point of expansion for $g(z_k)$. A reasonable choice is the MAP. Substitute $z_k = \tilde{z}_{ks}$ in (65) and use $\tilde{z}_{ks} = \hat{z}_{ks}$, we can obtain an iterative update for $\tilde{z}_{ks}$

$$\tilde{z}_{ks} \leftarrow \tilde{z}_{ks} + \eta\Lambda^{-1}\{R_k(y_k - g(\tilde{z}_{ks}))g'(\tilde{z}_{ks}) + G_{ks}(\zeta_{ks} - \tilde{z}_{ks})\}. \tag{67}$$

The η is the learning rate, and is introduced to avoid oscillation. This iterative update gives the signal log-spectral estimator, $\tilde{x}_{ks}$, which is the first element of the $\tilde{z}_{ks}$.

The state probability $p(s \mid y)$ is computed as, per Bayes' rule, $p(s \mid y) \propto p(y \mid s)p(s)$. The state-dependent probability is

$$p(y \mid s) = \prod_k \sqrt{\left|\frac{\Gamma_{ks}}{2\pi}\right|}\exp\left(-\frac{\Gamma_{ks}}{2}(y_k - l(\zeta_{ks}))^2\right) \tag{68}$$

where the mean $l(\zeta_{ks})$ is given in (64) and the precision $\Gamma_{ks} = (1/(g'^T G^{-1} g' + 1/R_k))$.

The log-spectral estimator is $\bar{x}_k = \sum_s \tilde{x}_{ks} p(s \mid y)$. Using the phase of the noisy signal $\angle Y_k$, the signal estimation in frequency domain is given by $X_k = \exp(\bar{x}_k/2)e^{i\angle Y_k}$.

It is observed that Newton's method with learning rate 1 oscillates; therefore, we set $\eta = 0.5$ in our experiments. We initialize the iteration of (67) with two conditions, $(y_k, \rho_k)^T$ and $(\mu_{ks}, \rho_k)^T$, and choose the one that offers higher likelihood value. The number of iterations is 7 which is enough for convergence. Note that the optimization of the two variables x and n increases computational cost.

4) Super Gaussian Prior (SuperGauss): This method is developed in [25]. Let $X_R = \mathrm{Re}\{X\}$ and $X_I = \mathrm{Im}\{X\}$ denote the real and the imaginary parts of the signal FFT coefficients. The super Gaussian priors for X_R and X_I obey double-sided exponential distribution, given by

$$p(X_R) = \frac{1}{\sigma_x}e^{-\frac{2|X_R|}{\sigma_x}} \tag{69}$$

$$p(X_I) = \frac{1}{\sigma_x}e^{-\frac{2|X_I|}{\sigma_x}}. \tag{70}$$

Assume the Gaussian density for the noise N, $p(N) = \mathcal{N}(0, 1/\sigma_n^2)$. Here, σ_x^2 and σ_n^2 are the means of $|X|^2$ and $|N|^2$, respectively. Let $\xi = \sigma_x^2/\sigma_n^2$ be the *a priori* SNR, $Y_R = \mathrm{Re}\{Y\}$ be the real part of the noisy signal FFT coefficient. Define $L_{R+} = 1/\sqrt{\xi} + Y_R/\sigma_n$, and $L_{R-} = 1/\sqrt{\xi} - Y_R/\sigma_n$. It was shown in [25, (11)] that the optimal estimator for the real part is

$$\hat{X}_R = Y_R + \frac{\sigma_n}{\sqrt{\xi}}\frac{e^{\frac{2Y_R}{\sigma_x}}\mathrm{erfc}(L_{R+}) - e^{-\frac{2Y_R}{\sigma_x}}\mathrm{erfc}(L_{R-})}{e^{\frac{2Y_R}{\sigma_x}}\mathrm{erfc}(L_{R+}) + e^{-\frac{2Y_R}{\sigma_x}}\mathrm{erfc}(L_{R-})} \tag{71}$$

where $\mathrm{erfc}(x)$ denotes the complementary error function. The optimal estimator for the imaginary part $\hat{X}_I$ is derived analogously in the same manner. The FFT coefficient estimator is given by $\hat{X} = \hat{X}_R + i\hat{X}_I$.

D. Comparison Criteria

The performance of the algorithms are subject to some quality measures. We employ three criteria to evaluate the performances of all algorithms: SNR, SD, and word recognition error rate. For all experiments, the estimated signal $\hat{x}[t]$ are normalized such that it has the same covariance as the clean signal $x[t]$ before computing the signal quality measures.

1) Signal-to-Noise Ratio (SNR): In time domain, SNR is defined by

$$\mathrm{SNR} = 10\log_{10}\frac{\sum_t (x[t])^2}{\sum_t(\hat{x}[t] - x[t])^2} \tag{72}$$

where $x[t]$ is original clean signal, and $\hat{x}[t]$ is estimated signal.

2) Spectral Distortion (SD): Let x_k^c and $\hat{x}_k^c$ be the cepstral coefficients of the clean signal and the estimated signal, respectively. The computation of cepstral coefficients is described in Section II-A. The spectral distortion is defined in [25] by

$$\mathrm{SD} = \sqrt{\sum_{k=1}^{16}(x_k^c - \hat{x}_k^c)^2} \tag{73}$$

where the first 16 cepstral coefficients are used.

3) Word Recognition Error Rate: We use the speech recognition engine provided on the ICSLP website [24]. The recognizer is based on the HTK package. The inputs of the recognizer include MFCC, its velocity (Δ MFCC) and its acceleration ($\Delta\Delta$ MFCC) that are extracted from speech waveforms. The words are modeled by the HMM with no skipover states and two states for each phoneme. The emission probability for each state is a GMM of 32 mixtures, of which the covariance matrices are diagonal. The grammar used in the recognizer is the same as the sentence grammar shown in Section V-A. More details about the recognition engine can be found at [24].

For each input SNR condition, the estimated signals are fed into the recognizer. A score of $\{0, 1, 2, 3\}$ is assigned to each utterance depending on how many key words (*color, letter, digit*) that are incorrectly recognized. The word recognition error rate in percentage is the average of the scores of all 600 testing utterances divided by 3.

E. Results

1) Performance Comparison With Fixed Gain and Known Noise Model: All the algorithms are applied to enhance the speech corrupted by SSN at various SNR levels. They are compared by SNR, SD, and word recognition error rate. The Wiener filer, which contains the strong and detailed signal prior from a clean speech, can be regarded as an experimental upper bound.

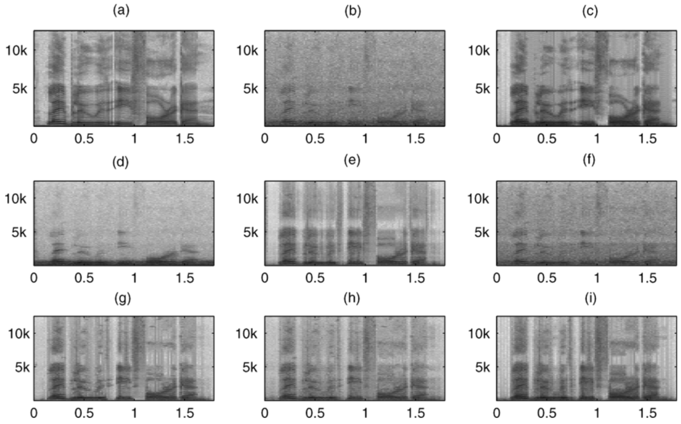


Fig. 5. Spectrogram of a female speech "lay blue with e four again." (a) Clean speech. (b) Noisy speech of 6-dB SNR. (c)–(i) Enhanced signals by (c) Wiener filter, (d) perceptual model (Wolfe), (e) linear approximation (Linear), (f) super Gaussian prior (SuperGauss), Laplace method in (g) frequency domain (LaplaceFFT) and in (h) log-spectral domain (LaplaceLS), (i) Gaussian approximation (Gaussian).

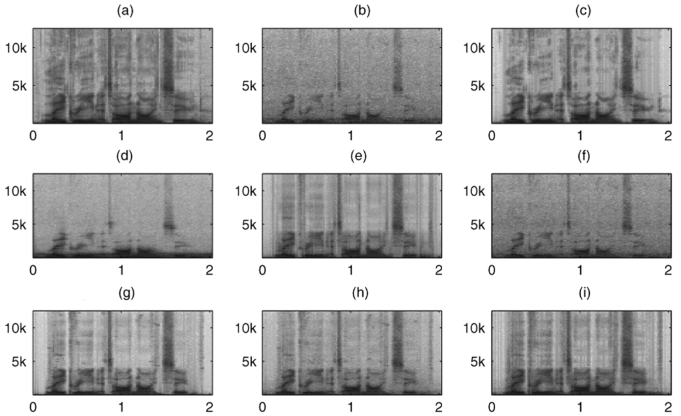
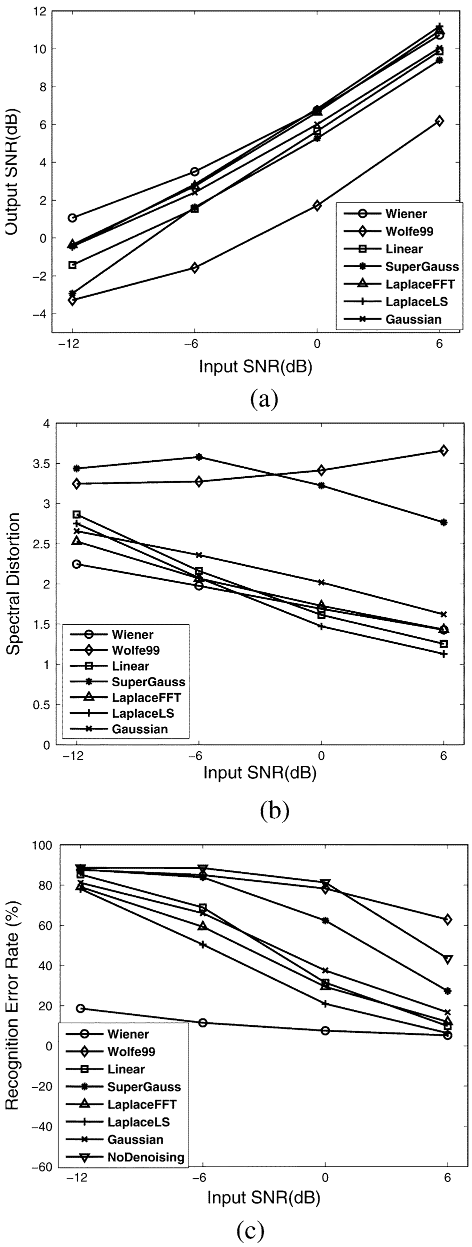


Fig. 6. Spectrogram of a male speech "lay green at r nine soon." (a) Clean speech. (b) Noisy speech of 6-dB SNR. (c)–(i) Enhanced signal by various algorithms. See Fig. 5. (a) Cleen Speech. (b) Noisy Speech. (c) Wiener Filter. (d) Wolfe. (e) Linear. (f) SuperGauss. (g) LaplaceFFT. (h) LaplaceLS. (i) Gaussian.



(a)

(b)

(c)

Fig. 7. Signal-to-noise ratio, spectrum distortion, and recognition error rate of speeches enhanced by the algorithms. The speech is corrupted at four input SNR values. The gain and the noise spectrum are assumed to be known. Wiener: Wiener filter; Wolfe99: perceptual model; Linear: linear approximation; SuperGauss: super Gaussian prior; LaplaceFFT: Laplace method in frequency domain; LaplaceLS: Laplace method in log-spectral domain; Gaussian: Gaussian approximation; NoDenoising: noisy speech input. (a) Signal-to-noise ratio. (b) Spectral distortion. (c) Recognition error rate.

Figs. 5 and 6 show the spectrograms of a female speech and a male speech, respectively. The SNR for the noisy speech is 6 dB. The Wiener filter can recover the spectrogram of the speech. The methods based on the models in log-spectral domain (Linear, LaplaceFFT, LaplaceLS, and Gaussian) can effectively suppress the SSN and recover the spectrogram. Because the SuperGauss estimates the real and imaginary parts separately, the spectral amplitude is not optimally estimated which leads to a blurred spectrogram. The perceptual model (Wofle99) fails to suppress SSN because of its spectral similarity to speech.

The SNR of speech enhanced by various algorithms are shown in Fig. 7(a). Wiener filter performs the best. Laplace methods (LaplaceFFT and LaplaceLS) are very effective, and the LaplaceLS is better. This coincides with the belief that the log-spectral amplitude estimator is more suitable for speech processing. The Gaussian approximation works comparably well to the Laplace methods with the advantage of greater computational efficiency where no iteration is necessary. The linear approximation provides inferior SNR. The reason is that this approach involves two hidden variables, which may increase the uncertainty for signal estimation. The SuperGauss works better than perceptual model (Wolfe99) which fails to suppress SSN.

The SD of speech enhanced by various algorithms are shown in Fig. 7(b). The methods that estimate spectral amplitude (Linear, LaplaceFFT, LaplaceLS) perform close to the Wiener filter. Because the SupperGauss estimates the real part and the imaginary part of FFT coefficients separately, it introduces distortion to the spectral amplitude and gives higher SD. The perceptual model is not effective to suppress SSN.

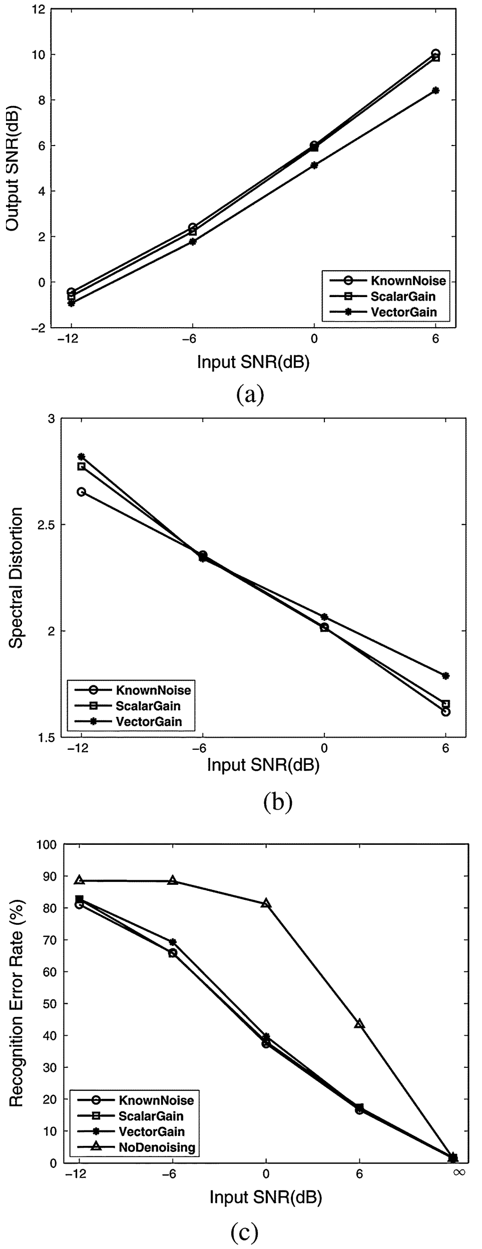
TABLE I
COMPUTATIONAL TIME (SECONDS) OF PROCESSING 10 S OF NOISY SPEECH SAMPLED AT 25 kHz

Wiener	**Linear**	**SupperGauss**
1.15 s	134 s	1.28 s
LaplaceFFT	**LaplaceLS**	**Gaussian**
57.3 s	40.5 s	0.25 s

The word recognition error rate of speech enhanced by various algorithms are shown in Fig. 7(c). The outstanding performance of Wiener filter may be considered as an upper bound. The Linear and LaplaceLS give very low word recognition error rate in the high SNR range, because they estimate the log-spectral amplitude, which is a strong fit to the recognizer input (MFCC). LaplaceLS is better than Linear in the low SNR range, because Linear involves two hidden variables to estimate. The LaplaceFFT and Gaussian also improve the recognition remarkably. Because SuperGauss offers less accurate spectral amplitude estimation and higher SD, it gives lower word recognition rate. The Wolfe99 is not able to suppress SSN and the decrease in performance may be caused by the spectral distortion.

The computation costs of these algorithms are given in Table I. All algorithms are implemented with MATLAB, and the experiments run on a 2.66-GHz PC. The methods based on linear approximation and Laplace method involve iterative optimization; thus, they are more computationally expensive. Their efficiency also depends on the number of initializations and iterations. The methods that do not involve iterations, Wiener filter, Gaussian, SuperGauss, are much faster.

2) Performance Comparison With Estimated Gain and Noise Spectrum: The performances of the Gaussian approximation with the fixed gain versus the estimated gain and noise spectrum are compared. The SNR, SD, and word recognition error rate of the enhanced speech are shown in Fig. 8(a)–(c), respectively. The performances are almost identical, which demonstrate that, under Gaussian approximation, the learning of gain and noise spectrum is very effective. Estimation of gain and noise degrades the performance compared to the scenario of fixed gain and known noise spectrum very slightly. Furthermore, with clean signal input, the estimated signal still has 32.71-dB SNR for scalar gain and 15.32-dB SNR for vector gain. The recognition error rate is also close to the results of the clean signal input. The slight degradation in the vector gain case is because we have more parameters to estimate.



(a)

(b)

(c)

Fig. 8. Signal-to-noise ratio, spectral distortion, and recognition error rate of speeches enhanced by algorithms based on Gaussian approximation. The speech is corrupted by SSN. KnownNoise: known gain and noise spectrum; ScalarGain: estimated frequency-independent gain and noise spectrum; VectorGain: estimated frequency dependent gain and noise spectrum; NoDenoising: noisy speech input. (a) Signal-to-noise ratio. (b) Spectral distortion. (c) Recognition error rate.

VI. CONCLUSION

We have developed speech enhancement algorithms based upon approximate Bayesian estimation. These approximations make the GMM in log-spectral domain applicable for speech enhancement. The log-spectral domain Laplace method, which computes the MAP estimator for the log-spectral amplitude, is particularly successful. It offers higher SNR, smaller recognition error rate, and lower SD. This confirms that the log-spectrum is more suitable for speech processing. The estimation of the log-spectral amplitude is a strong fit to the speech recognizer and significantly improves its performance, which makes this approach valuable to the recognition of the noisy speech. However, the Laplace method requires iterative optimization which increases the computational cost. Compared to the Laplace method, the Gaussian approximation with a closed-form signal estimation, is more efficient and performs comparably well. The advantage of fast gain and noise spectrum adaptation makes this algorithm more flexible. In the experiments, the proposed algorithms demonstrate superior performances over the spectral domain models and are able to reduce the noise effectively even when its spectral shape is similar to the speech.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for valuable comments which significantly improved the presentation.

REFERENCES

[1] Y. Ephraim and I. Cohen, "Recent advancements in speech enhancement," in *The Electrical Engineering Handbook*. Boca Raton, FL: CRC, 2006.

[2] H. Attias, J. C. Platt, A. Acero, and L. Deng, "Speech denoising and dereverberation using probabilistic models," in *Proc. NIPS*, 2000, pp. 758–764.

[3] S. Gannot, D. Burshtein, and E. Weinstein, "Signal enhancement using beamforming and nonstationarity with applications to speech," *IEEE Trans. Signal Process.*, vol. 49, no. 8, pp. 1614–1626, Aug. 2001.

[4] I. Cohen, S. Gannot, and B. Berdugo, "An integrated real-time beam-forming and postfiltering system for nonstationary noise environments," *EURASIP J. Appl. Signal Process.*, vol. 11, pp. 1064–1073, 2003.

[5] S. F. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-27, no. 2, pp. 113–120, Apr. 1979.

[6] Y. Ephraim and H. L. V. Trees, "A signal subspace approach for speech enhancement," *IEEE Trans. Speech Audio Process.*, vol. 3, no. 4, pp. 251–266, Jul. 1995.

[7] J. R. Hopgood and P. J. Rayner, "Single channel nonstationary stochastic signal separation using linear time-varying filters," *IEEE Trans. Signal Process.*, vol. 51, no. 7, pp. 1739–1752, Jul. 2003.

[8] A. Czyzewski and R. Krolikowski, "Noise reduction in audio signals based on the perceptual coding approach," in *Proc. IEEE WASPAA*, 1999, pp. 147–150.

[9] J.-H. Lee, H.-J. Jung, T.-W. Lee, and S.-Y. Lee, "Speech coding and noise reduction using ICA-based speech features," in *Proc. Workshop ICA*, 2000, pp. 417–422.

[10] P. Wolfe and S. Godsill, "Towards a perceptually optimal spectral amplitude estimator for audio signal enhancement," in *Proc. ICASSP*, 2000, vol. 2, pp. 821–824.

[11] Y. Ephraim, "Statistical-model-based speech enhancement systems," *Proc. IEEE*, vol. 80, no. 10, pp. 1526–1555, Oct. 1992.

[12] Y. Ephraim and D. Malah, "Speech enhancement using a minimum mean-square error short-time spectral amplitude estimator," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-32, no. 6, pp. 1109–1121, Dec. 1984.

[13] Y. Ephraim and D. Malah, "Speech enhancement using a minimum mean-square error log-spectral amplitude estimator," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-33, no. 2, pp. 443–445, Apr. 1985.

[14] Y. Ephraim, "A Bayesian estimation approach for speech enhancement using hidden Markov models," *IEEE Trans. Signal Process.*, vol. 40, no. 4, pp. 725–735, Apr. 1992.

[15] Y. Ephraim, "Gain-adapted hidden Markov models for recognition of clean and noisy speech," *IEEE Trans. Signal Process.*, vol. 40, no. 6, pp. 1303–1316, Jun, 1992.

[16] D. Burshtein and S. Gannot, "Speech enhancement using a mixture-maximum model," *IEEE Trans. Speech Audio Process.*, vol. 10, no. 6, pp. 341–351, Sep. 2002.

[17] B. Frey, T. Kristjansson, L. Deng, and A. Acero, "Learning dynamic noise models from noisy speech for robust speech recognition," in *Proc. NIPS*, 2001, pp. 1165–1171.

[18] T. Kristjansson and J. Hershey, "High resolution signal reconstruction," in *Proc. IEEE Workshop ASRU*, 2003, pp. 291–296.

[19] C. M. Bishop, *Neural Networks for Pattern Recognition*. New York: Oxford Univ. Press, 1995.

[20] H. Attias, "A variational Bayesian framework for graphical models," in *Proc. NIPS*, 2000, vol. 12, pp. 209–215.

[21] A. Azevedo-Filho and R. D. Shachter, "Laplace's method approximations for probabilistic inference in belief networks with continuous variables," in *Proc. UAI*, 1994, pp. 28–36.

[22] I. Cohen, "Noise spectrum estimation in adverse environments: Improved minima controlled recursive averaging," *IEEE Trans. Speech Audio Process.*, vol. 11, no. 5, pp. 466–475, Sep. 2003.

[23] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley-Interscience, 1991.

[24] M. Cooke and T.-W. Lee, "Speech separation challenge," [Online]. Available: http://www.dcs.shef.ac.uk/m̃artin/SpeechSeparationChallenge.html

[25] R. Martin, "Speech enhancement based on minimum mean-square error estimation and supergaussian priors," *IEEE Trans. Speech Audio Process.*, vol. 13, no. 5, pp. 845–856, Sep. 2005.

[26] P. Wolfe, "Example of short-time spectral attenuation," [Online]. Available: http://www.eecs.harvard.edu/p̃atrick/research/stsa.html

[27] I. Cohen and B. Berdugo, "Noise estimation by minima controlled recursive averaging for robust speech enhancement," *IEEE Signal Process. Lett.*, vol. 9, no. 1, pp. 12–15, Jan. 2002.

[28] R. McAulay and M. Malpass, "Speech enhancement using a soft-decision noise suppression filter," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-28, no. 2, pp. 137–145, Apr. 1980.

[29] R. Martin, "Noise power spectral density estimation based on optimal smoothing and minimum statistics," *IEEE Trans. Speech Audio Process.*, vol. 9, no. 5, pp. 504–512, Jul. 2001.

[30] D. Wang and J. Lim, "The unimportance of phase in speech enhancement," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. SP-30, no. 4, pp. 679–681, Apr. 1982.

[31] H. Attias, L. Deng, A. Acero, and J. Platt, "A new method for speech denoising and robust speech recognition using probabilistic models for clean speech and for noise," in *Proc. Eurospeech*, 2001, pp. 1903–1906.

[32] M. S. Brandstein, "On the use of explicit speech modeling in microphone array applications," in *Proc. ICASSP*, 1998, pp. 3613–3616.

[33] L. Hong, J. Rosca, and R. Balan, "Independent component analysis based single channel speech enhancement," in *Proc. ISSPIT*, 2003, pp. 522–525.

[34] C. Beaugeant and P. Scalart, "Speech enhancement using a minimum least-squares amplitude estimator," in *Proc. IWAENC*, 2001, pp. 191–194.

[35] T. Letter and P. Vary, "Noise reduction by maximum a posterior spectral amplitude estimation with supergaussian speech modeling," in *Proc. IWAENC*, 2003, pp. 83–86.

[36] C. Breithaupt and R. Martin, "MMSE estimation of magnitude-squared DFT coefficients with supergaussian priors," in *Proc. ICASSP*, 2003, pp. 848–851.

[37] J. Benesty, J. Chen, Y. Huang, and S. Doclo, "Study of the Wiener filter for noise reduction," in *Speech Enhancement*, J. Benesty, S. Makino, and J. Chen, Eds. new York: Springer, 2005.

Jiucang Hao received the B.S. degree from the University of Science and Technology of China (USTC), Hefei, and the M.S. degree from University of California at San Diego (UCSD), both in physics. He is currently pursuing the Ph.D. degree at UCSD.

His research interests are in developing new machine learning algorithms and applying them to areas such as speech enhancement, source separation, biomedical data analysis, etc.

Hagai Attias received the Ph.D. degree in theoretical physics from Yale University, New Haven, CT.

He is the President of Golden Metallic, Inc., San Francisco, CA. He has (co)authored over 60 scientific publications on machine learning theory and its applications in speech and audio processing, machine vision, and biomedical imaging. He has 12 issued patents. He was a Research Scientist at Microsoft Research, Redmond, WA, working in the Machine Learning and Applied Statistics Group. Several of his inventions at Microsoft were incorporated into the speech recognition engine used by the Windows operating system. Prior to that, he was a Sloan Postdoctoral Fellow at University of California, San Francisco (UCSF). At UCSF, he did some of the pioneering work on machine learning algorithms for audio analysis and source separation.

Srikantan Nagarajan received the M.S. and Ph.D. degrees in biomedical engineering from Case Western Reserve University, Cleveland, OH.

He did a Postdoctoral Fellowship at the Keck Center for Integrative Neuroscience, University of California, San Francisco (UCSF). Currently, he is a Professor in the Department of Radiology and Biomedical Imaging at UCSF and a faculty member in the UCSF/UCB Joint Graduate Program in Bioengineering. His research interests, in the area of neural engineering and machine learning, are to better understand neural mechanisms of sensorimotor learning and speech motor control, through the development of algorithms for improved functional brain imaging and biomedical signal processing.

Te-Won Lee received the M.S. degree and the Ph.D. degree (summa cum laude) in electrical engineering from the University of Technology Berlin, Berlin, Germany, in 1995 and 1997, respectively.

He was Chief Executive Officer and co-Founder of SoftMax, Inc., a start-up company in San Diego, CA, developing software for mobile devices. In December 2007, SoftMax was acquired by Qualcomm, Inc., the world leader in wireless communications where he is now a Senior Director of Technology leading the development of advanced voice signal processing technologies. Prior to Qualcomm and SoftMax, Dr. Lee was a Research Professor at the Institute for Neural Computation, University of California, San Diego, and a collaborating Professor in the Biosystems Department, Korea Advanced Institute of Science and Technology (KAIST). He was a Max-Planck Institute fellow (1995–1997) and a Research Associate at the Salk Institute for Biological Studies (1997–1999).

Dr. Lee received the Erwin-Stephan prize for excellent studies from the University of Technology Berlin and the Carl-Ramhauser prize for excellent dissertations from the Daimler–Chrysler Corporation. In 2007, he received the SPIE Conference Pioneer Award for work on independent component analysis and unsupervised learning algorithms.

Terrence J. Sejnowski (SM'91–F'06) is the Francis Crick Professor at The Salk Institute for Biological Studies, La Jolla, CA, where he directs the Computational Neurobiology Laboratory, an Investigator with the Howard Hughes Medical Institute, and a Professor of Biology and Computer Science and Engineering at the University of California, San Diego, where he is Director of the Institute for Neural Computation. The long-range goal of Dr. Sejnowski's laboratory is to understand the computational resources of brains and to build linking principles from brain to behavior using computational models. This goal is being pursued with a combination of theoretical and experimental approaches at several levels of investigation ranging from the biophysical level to the systems level. His laboratory has developed new methods for analyzing the sources for electrical and magnetic signals recorded from the scalp and hemodynamic signals from functional brain imaging by blind separation using independent components analysis (ICA). He has published over 300 scientific papers and 12 books, including *The Computational Brain* (MIT Press, 1994), with P. Churchland.

Dr. Sejnowski received the Wright Prize for Interdisciplinary research in 1996, the Hebb Prize from the International Neural Network Society in 1999, and the IEEE Neural Network Pioneer Award in 2002. His was elected an AAAS Fellow in 2006 and to the Institute of Medicine of the National Academies in 2008.