Abstract—We present a variational Bayesian algorithm for joint speech enhancement and speaker identification that makes use of speaker dependent speech priors. Our work is built on the intuition that speaker dependent priors would work better than priors that attempt to capture global speech properties. We derive an iterative algorithm that exchanges information between the speech enhancement and speaker identification tasks. With cleaner speech we are able to make better identification decisions and with the speaker dependent priors we are able to improve speech enhancement performance. We present experimental results using the TIMIT data set which confirm the speech enhancement performance of the algorithm by measuring signal-to-noise (SNR) ratio improvement and perceptual quality improvement via the PESQ score. We also demonstrate the ability of the algorithm to perform voice activity detection (VAD). The experimental results also demonstrate that speaker identification accuracy is improved.

Index Terms—Speech enhancement, speaker identification, variational Bayesian inference.

I. INTRODUCTION

Robust speaker recognition remains an important problem in statistical signal processing. Current approaches to speaker recognition mainly rely on directly modeling the speech feature vectors of the speakers to be identified and using clean speech to learn the parameters of these models. This approach makes these methods sensitive to noise and these systems do not perform well in real acoustic environments where noise is unavoidable. As a result the problem of robust speaker recognition continues to attract research interest (for example see [2]). Approaches include the use of robust features [3], [4] and feature compensation where speaker recognition features are post-processed to mitigate channel effects and noise [5]. Examples of this approach include cepstral mean subtraction (CMS) and RASTA speech processing [6]. Another approach involves the use of speech enhancement algorithms where the speech signal captured at the microphone is first enhanced to reduce the effects of noise and reverberation before speaker identification is performed.

Speech enhancement remains an active area of research (see [7] for a recent review). Speech enhancement algorithms can be broadly classified as spectral-subtractive, subspace or statistical-model based [7]. In spectral subtractive algorithms, an estimate of the noise spectrum is subtracted from the observed speech spectrum to obtain an estimate of the clean speech spectrum [8], [9]. Spectral subtractive algorithms are plagued by a number of drawbacks the most severe of which is the introduction of “musical” noise. Subspace algorithms rely on the decomposition of the noisy signal vector space into a speech signal subspace and a noise subspace and enhancing the observed signal by projecting it onto the speech signal subspace [10]. Similar ideas are present in the speaker recognition literature. For example in recent work by Kenny et al. [11], [12] the idea of eigen-voices is introduced which relies on the decomposition of the feature space into a subspace over which speaker variability is present and its orthogonal complement. Statistical-model based algorithms employ probabilistic models for both the speech and noise. The Ephraim-Malah enhancement algorithm [13] and its extensions [14], [15] provide excellent examples of statistical-model based algorithms. Here, the DFT coefficients of the clean speech and noise are assumed to be Gaussian distributed and a MMSE estimator for the spectral amplitude is derived. A major advantage of the Ephraim-Malah enhancement algorithm is that it does not suffer from the “musical noise” artifact [16]. In [17] the author derives a MMSE estimator for the spectral amplitude using the assumption that the spectral coefficients have super-Gaussian priors. In [18] the author proposes alternatives to the squared error distortion to derive perceptually motivated Bayesian estimators for the spectral amplitude starting with the assumption that the spectral coefficients of the clean speech are Gaussian distributed. In the papers discussed so far exact Bayesian inference is possible due to the assumption that certain parameters such as noise variances are known. Since these quantities are unknown in practice, speech enhancement would benefit from a full Bayesian treatment where these quantities are treated as unknown. For example, in this work we are able to infer SNR level from the observations making the algorithm robust to changes in noise level during the utterance.

A number of authors have presented speech enhancement algorithms which employ prior source models and approximate Bayesian methods (for example see [19], [20]). The Algonquin speech enhancement algorithm [21], [22] and some extensions [23], [24], [25], [26] apply a variational inference technique to enhance noisy reverberant speech using a speaker independent mixture of Gaussians speech prior in the log spectral domain. Our approach to robust speaker recognition is to use speaker...
dependent speech priors and to employ a Bayesian framework to estimate the clean speech and speaker identity jointly given an observed signal contaminated by additive noise [1], [27].

The Bayesian framework allows us to handle both parameter and model uncertainty in a principled way. Here, the parameters \( \theta \) and the observations \( \mathbf{X} \) are treated as random variables with a joint distribution \( p(\mathbf{X}, \theta) \). Given a particular joint distribution, we would like to compute the posterior distribution of the parameters given the observations in order to allow inference. Unfortunately, for most models of interest, including the model used in this paper, this posterior is intractable and we are forced to use approximations.

Variational inference methods have emerged as a powerful class of approximate inference techniques. In this approach inference is viewed as an optimization problem where an appropriate cost function is minimized [28]. Variational Bayesian inference [29] and modifications of belief propagation (BP) such as expectation propagation (EP)[30] fall in this category. The use of graphical models allows a powerful interpretation of variational techniques as message passing algorithms [31]. That is, the inference step consists of messages being passed between nodes in the graph with each node performing local computations. This allows the global inference problem to be decomposed into local computations [32].

Recently variational Bayesian methods have been successfully applied to several signal processing problems such as source separation [33] and parameter estimation [34] and to speech and language processing problems [35], [36], [37]. This provides motivation for the work presented here where variational Bayesian techniques are used to improve speaker recognition performance in noisy environments. In previous work we have considered the application of Markov chain Monte Carlo (MCMC) inference to the problem of joint enhancement and identification [27] and EP to joint source separation and identification [38]. The variational Bayesian approach offers advantages over both MCMC and EP. MCMC is computationally more expensive than VB making it less suitable for speech applications. Also, VB offers convergence guarantees that are lacking in EP.

The rest of the paper is organized as follows. In section II we present the problem formulation and characterize the joint distribution of the parameters and observations in our model. In section III we give a brief introduction to variational Bayesian inference and derive the joint speech enhancement and speaker identification algorithm by applying a variational approximation to the true posterior. Experimental results are presented in section IV. These results show that the proposed algorithm performs well in both speech enhancement and speaker identification. The algorithm outperforms the Ephraim-Malah algorithm [13], a standard baseline which has been found to outperform several speech enhancement algorithms in the literature [7, chapter 11], in both SNR improvement and perceptual quality as measured using the PESQ score. The ability of the algorithm to perform VAD is also experimentally verified. Section V presents a discussion and concludes the paper.

II. PROBLEM FORMULATION

In this work we use a source prior that takes into account the temporal correlation and non-gaussianity of speech. Using single channel observation of the noisy speech, the aim is to perform speech enhancement and speaker identification jointly.

We model speech as a time varying autoregressive (AR) process of order \( P \). For a given block \( k \) of speech samples \( s^k = [s^k_1, \ldots, s^k_N]^T \) we have (the speech signal is divided into \( K \) segments)

\[
s^k_n = \sum_{p=1}^{P} a^k_{p} s^k_{n-p} + \epsilon^k_n = (a^k)^T s^k_{n-1} + \epsilon^k_n \tag{1}
\]

where \( s^k_n = [s^k_n, \ldots, s^k_{n-P+1}]^T \), \( a^k = [a^k_1, \ldots, a^k_P]^T \) and \( \epsilon^k_n \sim \mathcal{N}(\epsilon^k_n; 0, (\tau^k_n)^{-1}) \). The signal observed at the microphone is given by

\[
r^k_n = s^k_n + \eta^k_n \tag{2}
\]

where \( \eta^k_n \sim \mathcal{N}(\eta^k_n; 0, (\tau^k_n)^{-1}) \) is additive white Gaussian noise with precision (inverse variance) \( \tau^k_n \).

From (1) we have

\[
p(s^k | a^k, \tau^k_n) = \prod_{n=1}^{N} p(s^k_n | s^k_{n-1}, a^k, \tau^k_n) = \prod_{n=1}^{N} \mathcal{N}(s^k_n; (a^k)^T s^k_{n-1}, (\tau^k_n)^{-1}). \tag{3}
\]

From (2) we can write \( p(r^k | s^k, \tau^k_n) = \mathcal{N}(r^k_n; s^k_n, \tau^k_n) \). If \( r^k = [r^k_1, \ldots, r^k_N]^T \) is the block of noisy observations corresponding to the source samples \( s^k \) the data likelihood is

\[
p(r^k | s^k, \tau^k_n) = \prod_{n=1}^{N} p(r^k_n | s^k_n, \tau^k_n) = \prod_{n=1}^{N} \mathcal{N}(r^k_n; s^k_n, \tau^k_n). \tag{4}
\]

To complete the probabilistic formulation we require priors over \( a^k \), \( \tau^k_n \), and \( \tau^k_n \). The speaker dependence is introduced by the prior over \( a^k \). We model the prior over \( a^k \) for speaker \( \ell \) as a Gaussian mixture model (GMM)

\[
p(a^k | \ell) = \sum_{m=1}^{M} \pi^\ell_m \mathcal{N}(a^k; \mu^\ell_m, \Sigma^\ell_m) \tag{5}
\]

where \( \ell \in \mathcal{L} = \{1, 2, \ldots, |\mathcal{L}|\} \) with \( \mathcal{L} \) being the library of known speakers. The parameters \( \{\mu^\ell_m, \Sigma^\ell_m, \pi^\ell_m\} \) for the distribution \( p(a^k | \ell) \) are obtained in advance from a corpus of clean speech.

We find it analytically convenient to introduce an indicator variable \( z^k_n \) that is a \( M_{\ell} |\mathcal{L}| \times 1 \) random binary vector that captures both the identity of the speaker and the mixture coefficient ‘active’ over a given frame. We have

\[
p(a^k | z^k_n) = \prod_{i=1}^{M_{\ell} |\mathcal{L}|} \left[ \mathcal{N}(a^k; \mu_i^\ell, \Sigma_i^\ell) \right]^{z^k_n,i}. \tag{6}
\]

The precisions \( \tau^k_n \) are assumed to have Gamma priors, that is

\[
p(\tau^k_n) = \text{Gam}(\tau^k_n; a, b),
\[
p(\tau^k_n) = \text{Gam}(\tau^k_n; a, b).
\]
The analytical forms of the Gaussian and Gamma distributions are presented in appendix A.

Now that we have the priors for all the random variables in our model we can write the joint distribution of the observations and parameters. We assume the joint distribution factors as shown in (7). We use the notation $x^{1:K}$ to denote the set $\{x_1, \ldots, x_K\}$.

$$ p(r^{1:K}, s^{1:K}, a^{1:K}, z_a^{1:K}, r^{1:K}, r^i_{\eta}) = \prod_k \left\{ p(x^k | s^k, r^k_\eta) \right\} \times p(s^k | a^k, r^k_\eta) p(a^k | z_a^k) p(r^k_\eta | \tau^k_\eta) \right\} p(z_a^{1:K}). \tag{7} $$

The prior $p(z_a^{1:K})$ is assumed to be factor as follows

$$ p(z_a^{1:K}) = p(z_a^1) \prod_{k=2}^K p(z_a^k | z_a^{k-1}). \tag{8} $$

This allows us to take into account the fact that adjacent speech blocks are likely to originate from the same speaker. In order to completely characterize (8) we need to know the speaker transition matrix $A = [a_{ij}]$ with $a_{ij} = p(\ell^k = i | \ell^{k-1} = j)$ where $\ell^k$ is the speaker responsible for the $k$th block and the mixture coefficients $\pi^k_\eta = [\pi^{\ell_1}_\eta, \ldots, \pi^{\ell_L}_\eta]_T$ for all the speakers in the library. The distribution $p(z_a^k | z_a^{k-1})$ is then characterized by the $M_a | \mathcal{L} \times M_a | \mathcal{L}$ matrix given by

$$ T = \begin{bmatrix} a_1 \otimes (\pi^1_\eta 1^T) \\ \vdots \\ a_L \otimes (\pi^L_\eta 1^T) \end{bmatrix} \tag{9} $$

where $a_\ell$ is the $\ell$th row of $A$. $1$ is a $M_a \times 1$ vector of all ones, and $\otimes$ represents the Kronecker product. We can now write

$$ p(z_a^k | z_a^{k-1}) = \prod_{i=1}^{M_a | \mathcal{L}} \prod_{j=1}^{M_a | \mathcal{L}} t_{ij}^{k-1} \sum_{r, a} \tag{10} $$

where $T = [t_{ij}]$. For compactness we represent all the parameters and latent variables as

$$ \Theta \equiv \{s^{1:K}, a^{1:K}, z_a^{1:K}, r^{1:K}, r^i_{\eta}^{1:K}\}. $$

Figure 1 shows a Bayesian network that captures the conditional dependencies between the random variables in our model.

Given the noisy observations, we would like to compute the posterior $p(z_a^k | r^{1:K})$ in order to determine the identity of the speaker responsible for generating the observed speech and the posterior $p(\theta^{1:K} | r^{1:K})$ in order to estimate the clean speech. However due to the intractability of these posteriors we employ approximate Bayesian inference techniques to compute them. The intractability results from the fact that we cannot compute expectations with respect to these posteriors.

![Bayesian network](image)

**III. VARIATIONAL BAYESIAN INFERENCE**

In variational Bayesian inference, we seek an approximation $q(\Theta)$ to the intractable posterior $p(\Theta | r^{1:K})$ which minimizes the Kullback-Leibler (KL) divergence between $q(\Theta)$ and $p(\Theta | r^{1:K})$ with $q(\Theta)$ constrained to lie within a tractable approximating family. The KL divergence $D(q || p)$ is a measure of the distance between two distributions and is defined by [39]

$$ D(q || p) = \int q(\Theta) \log \frac{q(\Theta)}{p(\Theta | r^{1:K})} d\Theta. $$

To ensure tractability we assume that the posterior can be written as a product of factors depending on disjoint subsets of $\Theta = \{\theta_1, \ldots, \theta_M\}$ [29], [40]. Assuming that each factor depends on a single element of $\Theta$ then

$$ q(\Theta) = \prod_{i=1}^M q_i(\theta_i). \tag{11} $$

It can be shown that the optimal form of $q_j(\theta_j)$ denoted by $q_j^*(\theta_j)$ that minimizes $D(q || p)$ is given by [40]

$$ \log q_j^*(\theta_j) = \mathbb{E}\{\log p(r^{1:K}, \Theta)\}_{q(\Theta \setminus \theta_j)} + \text{const.} \tag{12} $$

We use the notation $q(\Theta \setminus j)$ to denote the approximate posterior of all the elements of $\Theta$ except $\theta_j$. We obtain a set of coupled equations relating the optimal form of a given factor to the other factors. To solve these equations, we initialize all the factors and iteratively refine them one at a time using (12).

**A. Approximate Posterior**

Returning to the context of our joint speech enhancement and speaker ID model, we assume an approximate posterior $q(\Theta)$ that factorizes as follows

$$ q(\Theta) = \prod_k q(s^k) q(a^k) q(z_a^k) q(r^k_\eta) $$

The dependence of the posterior on the observations $r^{1:K}$ is implicit. Using (12) we obtain expressions for the optimal form of the factors. We obtain (see appendix B and C for details) 1)

$$ q^*(r^k_\eta) = \text{Gam}(r^k_\eta | a^*_\eta, b^*_\eta) \tag{13} $$

with

$$ a^*_\eta = \frac{a_\eta + N}{2}, $$

$$ b^*_\eta = b_\eta + \frac{1}{2} \mathbb{E}_s \{ \sum_{n=1}^N (r^k_n - s^k_n)^2 \}. $$


2) 

\[ q^*(\tau^k) = \text{Gam}(\tau^k|a^*_\tau, b^*_\tau) \]  

with 

\[ a^*_\tau = a_\tau + \frac{N}{2}, \]

\[ b^*_\tau = b_\tau + \frac{1}{2} \sum_{n=1}^{N} \left\{ \mathbb{E}\{s_n^k (s_n^k)^2\} - 2\mu_n^{a*}T\mathbb{E}\{s_n^ks_{n-1}^k\} \right\} \]

\[ + \mu_n^{a*}T\mathbb{E}\{s_{n-1}^ks_{n-1}^k\}\mu_n^{a*} + \text{Tr}(\mathbb{E}\{s_{n-1}^ks_{n-1}^k\}\Sigma_n^a) \} \]

Tr(.) is the trace of the matrix argument.

3) 

\[ q^*(\Sigma_n^a) = \prod_{i=1}^{M_a|L|} (\gamma^k_i)^{a^*_{1i}} \]  

where 

\[ \gamma^k_i = \frac{\rho^k_i}{\sum_{i=1}^{M_a|L|} \rho^k_i} \]

and 

\[ \log \rho^k_i = -\frac{1}{2} \log |\Sigma^a_i| - \frac{1}{2}(\mu^*_a - \mu^a_i)^T\Sigma^a_{i-1}(\mu^*_a - \mu^a_i) \]

\[ - \frac{1}{2} \text{Tr}(\Sigma^a_{i-1}\Sigma^*_a) + \sum_{j=1}^{M_a|L|} \gamma^k_j \log t_{ij} \]

\[ + \sum_{n=1}^{M_a|L|} \gamma^k_{n+1} \log t_{ni}. \]

Recall that \( t_{ij} \) are the elements of the matrix \( T \) introduced in section II.

4) 

\[ q^*(\alpha_n^k) = \mathcal{N}(\alpha_n^k|\mu^*_n, \Sigma^*_n) \]  

with 

\[ \Sigma^*_n = \left[ \sum_{n=1}^{N} \frac{a_n^*}{b_n^*} \mathbb{E}\{s_n^k s_{n-1}^k\} + \sum_{m=1}^{M_a|L|} \gamma^k_i \Sigma^a_{i-1} \right]^{-1} \]

\[ \mu^*_n = \Sigma^*_n \sum_{n=1}^{N} \frac{a_n^*}{b_n^*} \mathbb{E}\{s_n^k\} + \sum_{m=1}^{M_a|L|} \gamma^k_i \Sigma^a_{i-1} \mu^a_i \]

5) Turning to \( q^*(s^k) \) we have

\[ \log q^*(s^k) = -\frac{1}{2} \sum_{n=1}^{N} \frac{a_n^*}{b_n^*} (r_n^k - s_n^k)^2 \]

\[ - \frac{1}{2} \sum_{n=1}^{N} \frac{a_n^*}{b_n^*} ((s_n^k)^2 - 2\mu^*_ns_{n-1}^k) \]

\[ + \sum_{n=1}^{N} \frac{a_n^*}{b_n^*} \gamma^k_i s_n^k \]

\[ + \text{const}. \]  

As discussed in appendix B, \( \mathbb{E}\{s_n^k\} \), \( \mathbb{E}\{s_n^ks_{n-1}^k\} \) and \( \mathbb{E}\{s_n^ks_{n-1}^k\} \) can be computed using a Kalman smoother [41].

The forms of the expressions (13)-(16) are typical in Bayesian computations. They include a contribution from the prior and one from the data. The nature of the prior determines the relative contribution of the data component to the posterior. When the prior is uninformative, the posterior largely depends on the data.

B. The VB Algorithm

Armed with closed form expressions for the approximate forms of the posteriors for the parameters \( \alpha^k, \beta^k, \tau^k, \) and \( \tau_0^k \) and a means to compute the source statistics, we can now present the VB algorithm. The VB algorithm is similar to the expectation maximization (EM) algorithm. It consists of a step similar to the E-step where the current source estimates are determined using a Kalman smoother using the current estimates of the posterior parameters. In the VB-M step, the current source statistic estimates are used to update the parameters of the posterior distributions.

To run the algorithm, the noisy utterance is divided into \( K \) segments of \( N \) samples each. The posterior parameters for each block are initialized and updated at each iteration. See algorithm 1.

| Initialize the posterior distribution parameters \( \{a^*_n, b^*_n, \alpha^*_n, \Sigma^*_n, \gamma^*_n\} \) for all blocks; |
| for \( n = 1 \) to Number of Iterations do |
| \( \text{for } k = 1, \ldots, K \text{ do} \) |
| VB E-step: Run the Kalman smoother to estimate the source statistics for block \( k \); |
| VB M-Step: Update the posterior parameters for block \( k \) using (13)-(16); |
| end |
| end |

Algorithm 1: VB algorithm

IV. EXPERIMENTAL RESULTS

In this section we present experimental results that verify the performance of the algorithm. For the simulations we use the TIMIT database which contains recordings of 630 speakers drawn from 8 dialect regions across the USA with each speaker recording 10 sentences [42]. The sampling frequency of the utterances is 16kHz with 16 bit resolution. For our initial experiment a randomly generated library of four speakers was used. In order to train the speaker models we used 8 sentences and used the other 2 for testing. We assume an AR order of 8 with 10 mixture coefficients. To obtain training data for the AR models we divide the speech into 32ms frames and compute the AR coefficients corresponding to these frames using the Levinson-Durbin algorithm. We then use the EM algorithm to determine the GMM parameters. The EM algorithm is run until the relative change in model likelihood is less than 10^-4. 100 EM iterations are found to be sufficient. We also train speaker models using Mel Frequency Cepstral Coefficients (MFCCs) to allow us to compare the performance of our algorithm with that obtained using MFCCs. Here we use 13 coefficients obtained from 32ms frames with 50% overlap. Speaker GMMs are trained using the EM algorithm with the number of mixtures set at 32.

We found it necessary to augment the speaker library with a silence model to avoid erroneous classification of silent speech blocks. In our formulation, we treat ‘silence’ as an additional speaker therefore increasing the library size by one. The silence model consists of a single Gaussian with zero
and output SNRs are defined as clean, noisy and enhanced signals respectively, then the input measures the input and output SNR. If we assume that speaker changes can occur only after a silent state. That is (silence is considered the fifth speaker)

The experiments were performed using additive white Gaussian noise as the source of contamination. To run the algorithm, the noisy utterance was divided into 32ms segments \( N = 512 \). The hyperparameters of the gamma distributions were \( a = b = 10^{-6} \). Thus the prior over the noise variance is uninformative and the noise variance for a particular segment is inferred from the observation. This makes the algorithm robust to changes in noise level from segment to segment. As with any iterative algorithm, initialization is very important and it affects the quality of the final solution. In our experiments, the following initialization scheme was found to work well: We initialize the posterior mean of the AR coefficients to the AR coefficients obtained from the noisy speech blocks. The posterior covariance of the AR coefficients was initialized as the identity matrix. \( a^*_n \) and \( b^*_n \) are initialized to one for all blocks, \( b^*_n \) is initialized to the variance of the AR prediction error determined using the noisy speech block and \( a^*_n \) is initialized at one. Finally we initialize the parameters of \( q(z^k) \) as \( \gamma^k = \frac{1}{M_a|L|} \). The parameters of the transition matrix were set to \( p = q = 0.8 \). These values were determined by computing the transition probabilities between silence and speech states for several files from the TIMIT data set. The silence and speech states were determined using an energy detector.

Since we update the posterior parameters one at a time, we need to specify a parameter update schedule. The parameter update schedule is as follows:

1. Update the parameters of \( q^*(a^k) \).
2. Update the parameters of \( q^*(\tau^k) \).
3. Update the parameters of \( q^*(\tau^k) \).
4. Update the parameters of \( q^*(z^k) \).

This schedule was observed in simulation to be numerically stable.

To quantify the algorithm’s enhancement performance we measure the input and output SNR. If \( s, r \) and \( \hat{s} \) denote the clean, noisy and enhanced signals respectively, then the input and output SNRs are defined as

\[
\text{SNR}_{\text{in}} = 20 \log \frac{\|s\|}{\|s - r\|},
\]

\[
\text{SNR}_{\text{out}} = 20 \log \frac{\|\hat{s}\|}{\|\hat{s} - s\|}.
\]

In order to determine the appropriate number of iterations, we compute the average SNR improvement \( (\text{SNR}_{\text{out}} - \text{SNR}_{\text{in}}) \) after the final iteration of the algorithm for all the test utterances in the library for various values of number of iterations. Figure 2 shows a plot of SNR improvement versus number of iterations for two values of input SNR: 5 and 10dB. We see that there is minimal SNR improvement after 10 iterations. However, we set the number of iterations at 30 since this is observed to improve speaker identification performance. Figure 3 shows the spectrograms and speech waveforms corresponding to the utterance “The shot reverberated in diminishing whiplashes of sound” when corrupted by additive white Gaussian noise at 10dB and enhanced using the algorithm. Using a C implementation of the algorithm we can process a 3 second utterance in approximately 10 seconds when the algorithm is run for 10 iterations. A C implementation of the Ephraim-Malah enhancement algorithm processes the same utterance in less than one second.

![Fig. 2. SNR improvement (SNR_{out} − SNR_{in}) after the final iteration of the algorithm versus number of iterations.](image-url)
libraries of four speakers drawn from the TIMIT database. We performed experiments to investigate the average SNR improvement and speaker recognition rates as a function of input SNR. The algorithm was run for 30 iterations. Figure 4(a) shows a plot of the SNR improvement versus input SNR while figure 4(b) shows the recognition rates averaged over 100 random sets of four speakers each. We compare the SNR improvement of the algorithm to the SNR improvement obtained using the Ephraim-Malah enhancement algorithm [13] and using a Kalman smoother when the true AR coefficients are assumed known. That is, we obtain the AR coefficients from the clean speech and use these ARs to enhance the noisy speech using a Kalman smoother. The latter provides an upper bound to the performance of the algorithm since we employ a Kalman smoother in the VB E-step to enhance the noisy speech using the current estimate of the AR coefficients. Since we are working with an estimate of the AR coefficients obtained from noisy observations, we can not outperform the SNR improvement obtained by a Kalman smoother using the true AR coefficients. We also compare the recognition rates of the algorithm to those obtained when 1) AR coefficients are obtained directly from the noisy signals 2) MFCCs are obtained from the noisy signal 3) MFCCs are obtained from the VB enhanced signal and 4) MFCCs are obtained from the Ephraim-Malah enhanced signal.

From these results we see that significant SNR improvement is obtained by the algorithm with a maximum SNR improve-
ment of approximately 10dB obtained when the input SNR is -5dB. The VB algorithm outperforms Ephraim-Malah when the input SNR is between -5 and 7.5 dB. When the input SNR is between -5dB and 5dB, the SNR improvement obtained by the VB algorithm is within 1 dB of the performance obtained when the true AR coefficients are known (the upper bound since we have to estimate the AR coefficients and can not outperform a method in which these coefficients are known). Turning to speaker identification results, we see that the VB algorithm which relies on AR coefficients achieves performance comparable to MFCCs obtained directly from the noisy speech. We see that the best identification rates are obtained when MFCCs obtained using the enhanced speech are used. The MFCCs obtained from speech enhanced using the VB algorithm outperform MFCCs from speech enhanced using the Ephraim-Malah algorithm by up to approximately 5%. This shows that the improved performance of the VB algorithm in speech enhancement allows for improved speaker identification.

We are also interested in the perceptual quality of the speech enhanced using our algorithm. To this end we evaluate the Perceptual Evaluation of Speech Quality (PESQ) score of the enhanced utterances. The PESQ score is highly correlated to the mean opinion score (MOS) which is a subjective measure of speech quality [46]. To evaluate the MOS, listeners are asked to rate speech quality on a scale ranging from 1 to 5 with 1 being the worst and 5 the best [7]. In our experiments 80 files corrupted at input SNRs ranging from 0-10 dB were enhanced using both our algorithm and Ephraim-Malah. For each file we compute both the input and output PESQ score. Figure 5 shows the PESQ scores for both the VB algorithm and Ephraim-Malah and the best-fit lines. We see that the VB algorithm outperforms the Ephraim-Malah algorithm in terms of perceptual quality.

![Fig. 5. Comparison of perceptual quality performance between the VB algorithm and Ephraim-Malah](image)

In order to evaluate the performance of the VB algorithm in more realistic noisy conditions, experiments were performed using the NOIZEUS data set [7]. This data set contains 30 IEEE sentences corrupted by real world noises at various SNRs. The SNR improvement obtained by the VB algorithm is compared to that obtained using the Ephraim-Malah algorithm. Table I presents the average SNR improvement for all 30 sentences in the data set at input SNRs ranging from 0dB to 15dB. From the experimental results we see that the VB algorithm outperforms the Ephraim-Malah algorithm in the input SNR range 5dB to 15dB. However at 15dB, both algorithms introduce distortion leading to degradation of the signal.

<table>
<thead>
<tr>
<th>Noise Type</th>
<th>Algorithm</th>
<th>Input SNR (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ephraim-Malah</td>
<td>0</td>
</tr>
<tr>
<td>Train</td>
<td>VB</td>
<td>2.41</td>
</tr>
<tr>
<td></td>
<td>Ephraim-Malah</td>
<td>3.07</td>
</tr>
<tr>
<td>Airport</td>
<td>VB</td>
<td>1.10</td>
</tr>
<tr>
<td></td>
<td>Ephraim-Malah</td>
<td>1.94</td>
</tr>
<tr>
<td>Car</td>
<td>VB</td>
<td>1.82</td>
</tr>
<tr>
<td></td>
<td>Ephraim-Malah</td>
<td>5.14</td>
</tr>
</tbody>
</table>

We now present experimental results that demonstrate the algorithm’s performance in voice activity detection (VAD). All blocks assigned to the ‘silence’ speaker are classified as silence while blocks assigned to other speakers in the library are collectively classified as ‘speech’. Figures 6-7 show the VAD decisions obtained by the VB algorithm and the ITU-G.729 algorithm [45] when the speech is corrupted by additive white Gaussian noise at 10dB and -5dB. We compare the VAD decisions to the ground truth. To obtain the ground truth we perform energy thresholding on the clean speech. Any blocks with energy 20dB lower than the maximum energy are classified as silence. To quantify VAD performance, we compare the percentage of speech samples correctly identified as either silence or speech by the VB algorithm and the ITU-G.729 algorithm. Table II presents the experimental results when 80 speech files were processed at SNRs ranging from -5dB to 10dB by the two algorithms. We see that the VB algorithm outperforms the ITU-G.729 algorithm at all input SNRs considered.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Input SNR (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>VB</td>
<td>59.9</td>
</tr>
<tr>
<td>ITU-G.729</td>
<td>51.1</td>
</tr>
</tbody>
</table>

V. DISCUSSION AND CONCLUSIONS

Experimental results reported in the previous section verify that the proposed VB algorithm does indeed perform joint speech enhancement and speaker identification. The significant SNR improvement of up to 10dB obtained by the VB algorithm over a wide range of input SNRs shows that speech enhancement is achieved. Furthermore, when the input SNR is between -5dB and 5dB, the SNR improvement obtained by
the VB algorithm is within 1 dB of the upper bound obtained when the true AR coefficients are known. The enhancement performance is also confirmed by observing the time domain speech plots and spectrograms in figure 3 and by informal listening tests. Also, the VB algorithm outperforms the Ephraim-Malah algorithm, a standard baseline which has been found to outperform several speech enhancement algorithms in the literature [7, chapter 11], in terms of SNR improvement and perceptual quality as measured using the PESQ score. This result suggests that the full Bayesian treatment employed in the VB algorithm improves speech enhancement performance when compared to an algorithm in which some parameters are assumed known as is the case with the Ephraim-Malah algorithm. In the identification experiments, MFCCs from speech enhanced using the VB algorithm outperform MFCCs from speech enhanced using the Ephraim-Malah algorithm in the input SNR range of -5dB to 10dB. As an added benefit, the VB algorithm allows us to perform VAD. From the experimental results, we see that the VB algorithm outperforms the ITU-G.729 algorithm [45].

In this paper we have presented a variational Bayesian algorithm that performs speech enhancement and speaker identification jointly. We demonstrate the power of approximate Bayesian methods when applied to complex inference problems. The importance of considering speech enhancement and speaker identification jointly within a Bayesian framework is that we can use rich speaker dependent speech priors to mitigate the effects of noise and therefore improve speaker identification in noisy environments. The experimental results provided verify the performance of the algorithm.

APPENDIX A

STANDARD DISTRIBUTIONS

For an $N$-dimensional Gaussian random vector, we have

$$\mathcal{N}(x; \mu, \Sigma) = \frac{1}{2\pi^{N/2} |\Sigma|^{1/2}} \exp\left\{-\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu)\right\}$$

where $\mu$ is the $N$-dimensional mean vector and $\Sigma$ is the $N \times N$ covariance matrix.

The Gamma distribution over a positive random variable $\tau$ is given by

$$\text{Gam}(\tau; a, b) = \frac{1}{\Gamma(a)} b^a \tau^{a-1} \exp^{-b\tau}.$$  

APPENDIX B

APPROXIMATE POSTERIOR DERIVATIONS

In this appendix we derive the optimal factors of the approximate posterior presented in section III-A. Starting with the optimal form of $q(\tau^k_n)$ we have

$$\log q^*(\tau^k_n) = \mathbb{E}_{\Theta \setminus \tau^k_n} \{\log p(x^{1:K}, \Theta)\} + \text{const.}$$

$$= \mathbb{E}_{a^*} \{\log p(x^{k}|s^k_n, \tau^k_n)\} + \log p(\tau^k_n) + \text{const.}$$

$$= \mathbb{E}_{a^*} \left\{ \sum_{n=1}^N \log \mathcal{N}(r^k_n; s^k_n, \tau^k_n) \right\} + \log p(\tau^k_n) + \text{const.}$$

$$= \mathbb{E}_{a^*} \left\{ \frac{1}{2} \log \tau^k_n - \frac{\tau^k_n}{2} (r^k_n - s^k_n)^2 \right\} + (a_n - 1) \log \tau^k_n + \text{const.}$$

$$= (a_n + \frac{N}{2} - 1) \log \tau^k_n - \tau^k_n |b_n + \frac{1}{2} \mathbb{E}_{a^*} \{ \sum_{n=1}^N (r^k_n - s^k_n)^2 \} | + \text{const.} \quad (19)$$

From (19) we obtain (13)

$$q^*(\tau^k_n) = \text{Gam}(\tau^k_n|a^*_n, b^*_n)$$

with

$$a^*_n = a_n + \frac{N}{2},$$

$$b^*_n = b_n + \frac{1}{2} \mathbb{E}_{a^*} \{ \sum_{n=1}^N (r^k_n - s^k_n)^2 \}.$$
For \( q(\tau^k) \) we have

\[
\log q^*(\tau^k) = \mathbb{E}_{\Theta \setminus \tau^k} \{ \log p(\Theta^1; K, \Theta) \} + \text{const.}
\]

\[
= \mathbb{E}_{\Theta^k, \mu^k} \{ \log p(s^k | \mu^k, \tau^k) \} + \log p(\tau^k) + \text{const.}
\]

\[
= \mathbb{E}_{\Theta^k, \mu^k} \left\{ \sum_{n=1}^{N} \log \mathcal{N}(s^k_n; a^k T s^k_{n-1}, (\tau^k)^{-1}) \right\}
\]

\[
+ \log p(\tau^k) + \text{const.}
\]

\[
= \mathbb{E}_{\Theta^k, \mu^k} \left\{ \sum_{n=1}^{N} \left( \frac{1}{2} \log \tau^k_n - \frac{\tau^k_n}{2} (s^k_n - a^k T s^k_{n-1})^2 \right) \right\}
\]

\[
+ (a_c - 1) \log \tau^k_c - b_c \tau^k_c + \text{const.}
\]

From (20) we obtain (14)

\[
q^*(\tau^k) = \text{Gam}(\tau^k_c | a^*_c, b^*_c)
\]

with

\[
a^*_c = a_c + \frac{N}{2},
\]

\[
b^*_c = b_c + \frac{1}{2} \mathbb{E}_{\Theta^k, \mu^k} \left\{ \sum_{n=1}^{N} (s^k_n - a^k T s^k_{n-1})^2 \right\}.
\]

Turning to \( q(z^k) \) we have

\[
\log q^*(z^k) = \mathbb{E}_{\Theta \setminus z^k} \{ \log p(\Theta^1; K, \Theta) \} + \text{const.}
\]

\[
= \mathbb{E}_{\Theta^k} \{ \log p(a^k | z^k) \} + \mathbb{E}_{z_{a,i}} \{ \log p(z_{a,i} | z_{a,i}^{-1}) \}
\]

\[
+ \mathbb{E}_{z_{a,i}} \left\{ \sum_{n=1}^{N} (z_{a,i}^k - a^k T z_{a,i}^{k-1})^2 \right\} + \text{const.}
\]

Considering \( q(a^k) \) we have

\[
\log q^*(a^k) = \mathbb{E}_{\Theta \setminus a^k} \{ \log p(\Theta^1; K, \Theta) \} + \text{const.}
\]

\[
= \mathbb{E}_{\tau^k} \{ \log p(\tau^k) \} + \mathbb{E}_{\tau^k} \{ \log p(\tau^k) \} + \text{const.}
\]

\[
= \mathbb{E}_{\tau^k} \left\{ \sum_{n=1}^{N} \log \mathcal{N}(s^k_n; a^k T s^k_{n-1}, (\tau^k)^{-1}) \right\}
\]

\[
+ \mathbb{E}_{\tau^k} \left\{ \sum_{n=1}^{N} (z_{a,i}^k - a^k T z_{a,i}^{k-1})^2 \right\} + \text{const.}
\]

\[
= \frac{1}{2} \sum_{n=1}^{N} \mathbb{E}_{\tau^k} \{ (a^k - \mu^k)^T \Sigma_i^{-1} (a^k - \mu^k) \}
\]

\[
+ \text{const.}
\]

(22) is quadratic in \( a^k \) and we can write

\[
\log q^*(a^k) = -\frac{1}{2} a^k T \left[ \sum_{n=1}^{N} \mathbb{E}_{\tau^k} \{ s^k_n s^k_{n-1} T \} \right] a^k
\]

\[
+ \sum_{i=1}^{M_a | \mathcal{L}} \mathbb{E}_{\tau^k} \{ z_{a,i}^k \} \frac{1}{2} \log |\Sigma_i^a|
\]

\[
+ \sum_{i=1}^{M_a | \mathcal{L}} \mathbb{E}_{\tau^k, \mu^k} \{ z_{a,i}^k \} \{ (a^k - \mu^k)^T |\Sigma_i^a^{-1} (a^k - \mu^k) \}
\]

\[
+ \text{const.}
\]

From (23) we obtain (16)

\[
q^*(a^k) = \mathcal{N}(a^k; \mu_a^*, \Sigma_a^*)
\]

with

\[
\Sigma_a^* = \left[ \sum_{n=1}^{N} \mathbb{E}_{\tau^k \setminus \tau^k_c} \{ s^k_n s^k_{n-1} T \} \right]^{-1}
\]

\[
\mu_a^* = \Sigma_a \left[ \sum_{n=1}^{N} \mathbb{E}_{\tau^k \setminus \tau^k_c} \{ s^k_n s^k_{n-1} \} \right]
\]

\[
+ \sum_{i=1}^{M_a | \mathcal{L}} \mathbb{E}_{\tau^k \setminus \tau^k_c} \{ z_{a,i}^k \} \frac{1}{2} \log |\Sigma_i^a|
\]

\[
+ \sum_{i=1}^{M_a | \mathcal{L}} \mathbb{E}_{\tau^k \setminus \tau^k_c} \{ z_{a,i}^k \} \{ (a^k - \mu_i^k)^T \Sigma_i^a^{-1} (a^k - \mu_i^k) \}
\]

\[
+ \text{const.}
\]
Turning to \( q^*(s^k) \) we have

\[
\log q^*(s^k) = \mathbb{E}_{\Theta | y^k, \Theta} \{ \log p(r^{1:K}, \Theta) \} + \text{const.}
\]

\[
= \mathbb{E}_{\Theta | y^k} \{ \log p(r^k | s^k, \tau^k_1, \tau^k_2) \} + \mathbb{E}_{\Theta | y^k} \{ \log p(s^k | a^k, \tau^k_1, \tau^k_2) \} + \text{const.}
\]

\[
= \mathbb{E}_{\Theta | y^k} \left\{ \sum_{n=1}^{N} \log \mathcal{N}(r^k_n; s^k_n, \tau^k_1) \right\} + \mathbb{E}_{\Theta | y^k} \left\{ \sum_{n=1}^{N} \log \mathcal{N}(s^k_n; a^{kT} s^k_{n-1}, (\tau^k_2)^{-1}) \right\} + \text{const.}
\]

\[
= \mathbb{E}_{\Theta | y^k} \left\{ \sum_{n=1}^{N} \left[ -\frac{\tau^k_k}{2} (r^k_n - s^k_n)^2 \right] \right\} + \text{const.}
\]

\[
= \mathbb{E}_{\Theta | y^k} \left\{ \sum_{n=1}^{N} \left[ -\frac{\tau^k_k}{2} (s^k_n - a^{kT} s^k_{n-1})^2 \right] \right\} + \text{const.}
\]

(24)

Expanding the terms in (24) and evaluating the expectations yields (17).

\[
\log q^*(s^k) = -\frac{1}{2} \sum_{n=1}^{N} \frac{\alpha^*_{n}}{b^*_{n}} (r^k_n - s^k_n)^2
\]

\[
- \frac{1}{2} \sum_{n=1}^{N} \frac{\alpha^*_{n}}{b^*_{n}} \left( (s^k_n)^2 - 2 \mu^*_{n} s^k_n s^k_{n-1} - s^k_{n-1} \mu^*_{n} s^k_n + s^k_{n-1} \Sigma^*_{n-1} s^k_{n-1} \right) + \text{const.}
\]

To arrive at the conclusion that \( \mathbb{E}\{s^k_n\}, \mathbb{E}\{s^k_n s^{kT}_{n-1}\} \) and \( \mathbb{E}\{s^k_n s^{kT}_{n-1}\} \) can be computed using a Kalman smoother consider the following state space model where \( y^k = [r^k_n, 0, \ldots, 0]^T \)

\[
s^k_n = A s^k_{n-1} + G u^k_n
\]

\[
y^k_n = H s^k_n + v^k_n
\]

(25)

(26)

with

\[
u^k \sim \mathcal{N}(u^k; 0, (\tau^k_2)^{-1})
\]

\[
v^k \sim \mathcal{N}(v^k; 0, \Sigma^k_v)
\]

(27)

(28)

where

\[
A = \begin{pmatrix}
\mu^*_{1,a} & \mu^*_{2,a} & \ldots & \mu^*_{p,a} \\
1 & 0 & \ldots & 0 \\
0 & 1 & \ldots & 0 \\
\vdots & \ddots & \vdots & \vdots \\
0 & \ldots & 1 & 0
\end{pmatrix}
\]

\[
G = \begin{bmatrix}
1 & 0 & \ldots & 0
\end{bmatrix}^T
\]

(29)

(30)

and

\[
H = \begin{bmatrix}
1, 0, \ldots, 0
\end{bmatrix}^T
\]

(31)

Also

\[
\Sigma^k_v = \begin{bmatrix}
(\tau^k_2)^{-1} & \tau^k_2 (\Sigma^*_{n-1})^{-1}
\end{bmatrix}
\]

(32)

Consider the sequence of observations \( \{y^k_1, \ldots, y^k_N\} \) and the corresponding states \( \{s^k_1, \ldots, s^k_N\} \). The joint distribution for the state space model is

\[
p(y^k_1, \ldots, y^k_N, s^k_1, \ldots, s^k_N) = \prod_{n=1}^{N} p(y^k_n | s^k_n) p(s^k_1 | s^k_{n-1})
\]

\[
= \prod_{n=1}^{N} p(y^k_n | s^k_n) p(s^k_1 | s^k_{n-1}).
\]

The posterior

\[
p(s^k_1, \ldots, s^k_N | y^k_1, \ldots, y^k_N) \propto p(y^k_1, \ldots, y^k_N, s^k_1, \ldots, s^k_N)
\]

and

\[
\log p(s^k_1, \ldots, s^k_N | y^k_1, \ldots, y^k_N) = \sum_{n=1}^{N} \log p(y^k_n | s^k_n)
\]

\[
+ \sum_{n=1}^{N} \log p(s^k_n | s^k_{n-1}) + \text{const.}
\]

(33)

From (25) to (28) we can write

\[
p(y^k_n | s^k_n) = \mathcal{N}(y^k_n; H s^k_n, \Sigma^k_v)
\]

\[
p(s^k_1 | s^k_{n-1}) = \mathcal{N}(s^k_1; \mu^*_{n} s^k_n + s^k_{n-1} \Sigma^*_{n-1}, (\tau^k_2)^{-1})
\]

And evaluating (33) we obtain

\[
\log p(s^k_1, \ldots, s^k_N | y^k_1, \ldots, y^k_N) = -\frac{1}{2} \sum_{n=1}^{N} (s^k_n - \mu^*_{n} s^k_n)^2
\]

\[
- \frac{1}{2} \sum_{n=1}^{N} \left( \frac{\tau^k_k}{2} (s^k_n - s^k_{n-1})^2 + \tau^k_2 (s^k_n - s^k_{n-1}) (s^k_n - s^k_{n-1}) \right) + \text{const.}
\]

(34)

Comparing (17) and (34) we see that the two expressions are equivalent and we conclude that we can compute \( \mathbb{E}\{s^k_n\}, \mathbb{E}\{s^k_n s^{kT}_{n-1}\} \) and \( \mathbb{E}\{s^k_n s^{kT}_{n-1}\} \) using a Kalman smoother if we assume that the observations are generated by the state space model described by (25) to (28). We have \( \mathbb{E}\{s^k_n\} = \mathbb{E}\{s^k_n, s^{kT}_{n-1}\} \) and the quantity \( \mathbb{E}\{s^k_n\} \) is obtained from the posterior means computed by the Kalman smoother. Also \( \mathbb{E}\{s^k_n s^{kT}_{n-1}\} = \text{Cov}(s^k_n) + \mathbb{E}\{s^k_n\} \mathbb{E}\{s^{kT}_{n-1}\} \) and \( \text{Cov}(s^k_n) \) is obtained from the Kalman smoother and the second order moments \( \mathbb{E}\{(s^k_n)^2\} \) are obtained as follows

\[
\mathbb{E}\{(s^k_n)^2\} = \mathbb{E}\{s^k_n s^{kT}_{n-1}\}, 1, 1.
\]

Similarly \( \mathbb{E}\{s^k_n s^{kT}_{n-1}\} = \text{Cov}(s^k_n) + \mathbb{E}\{s^k_n\} \mathbb{E}\{s^{kT}_{n-1}\} \) and \( \text{Cov}(s^k_n) \) is obtained from the Kalman smoother and \( \mathbb{E}\{s^k_n s^{kT}_{n-1}\} \) is obtained from the first row of \( \mathbb{E}\{s^k_n s^{kT}_{n-1}\} \).

APPENDIX C

REQUIRED EXPECTATIONS

To characterize the parameters of the posterior distributions derived in appendix B we need to compute the following expectations:
The first and second order moments $\mathbb{E}\{s_n^2\}$ and $\mathbb{E}\{(s_n^*)^2\}$ are computed using a Kalman smoother as discussed in appendix B.

\[
\mathbb{E}_k\left\{ \sum_{n=1}^{N} (r_n - s_n)^2 \right\} = \mathbb{E}_k\left\{ \sum_{n=1}^{N} (r_n^2 - 2r_n^* s_n + (s_n^*)^2) \right\}
\]

\[
\mathbb{E}_k\left\{ \sum_{n=1}^{N} \left( k_n - a^T k_n^{-1} \right)^2 \right\} = \mathbb{E}_k\left\{ \sum_{n=1}^{N} \left( k_n - a^T k_n^{-1} \right)^2 \right\} + \mathbb{E}_k\left\{ \sum_{n=1}^{N} \right. \left( \mu_n^* - \mu_n \right) (\mu_n^* - \mu_n) + \text{Tr}(\Sigma_n^{-1} \Sigma_n)
\]

\[
\mathbb{E}_k\left\{ \sum_{n=1}^{N} \left( k_n^* - \mu_n^* \right) \right\} = \sum_{n=1}^{N} \mathbb{E}_k\left\{ \sum_{n=1}^{N} \left( k_n^* - \mu_n^* \right) \right\} + \text{Tr}(\Sigma_n^{-1} \Sigma_n)
\]

\[
\mathbb{E}_k\left\{ \sum_{n=1}^{N} \left( k_n - \mu_n \right) \right\} = \gamma_k
\]

REFERENCES


Ciira wa Maina (S’08) received the B.Sc. degree (First class honours) in electrical engineering from the University of Nairobi, Kenya in 2007. Since September 2007 he has been pursuing a Ph.D. at Drexel University, Philadelphia, PA. His research interests include statistical signal processing, speech processing, approximate Bayesian inference and inference in graphical models.

John M. Walsh (S’01-M’07) received the B.S. *(magna cum laude)*, M.S. and Ph.D. degrees in electrical and computer engineering from Cornell University, Ithaca, NY in 2002, 2004, and 2006, respectively.

In September 2006, he joined the Department of Electrical and Computer Engineering at Drexel University, Philadelphia, PA, where he is currently an assistant professor. His current research interests include: (a) the performance and convergence of distributed collaborative estimation in wireless sensor networks via expectation propagation, (b) delay mitigating codes and rate-delay tradeoffs in multipath routed and network coded networks, and (c) joint source separation and identification.

Dr. Walsh is a member of HKN and TBP.