AN MCMC APPROACH TO JOINT ESTIMATION OF CLEAN SPEECH AND NOISE FOR ROBUST SPEECH RECOGNITION

Aleem Mushtaq and Chin-Hui Lee

School of ECE, Georgia Institute of Technology, Atlanta, GA, 30332-0250, USA

ABSTRACT

We present a novel framework for joint estimation of speech and noise statistics using a Markov chain Monte Carlo approximation. The underlying distributions of the speech and noise components of noisy speech are estimated at each frame and inferences are made from these distributions. The clean speech is approximated by a discrete distribution, from which new features are extracted and used in the recognition process. The availability of information about the noise statistics enables the algorithm to handle non-stationary noise within an utterance and also improves the overall recognition performance when compared to the previously available sequential Monte Carlo (particle filter) methods for noisy speech compensation. We report experimental results obtained with the Aurora-2 connected digit recognition task and achieve an error reduction of 12.87% over state-of-the-art multi-condition training.

Index Terms— Monte Carlo, Markov chain Monte Carlo, robust speech recognition, noise compensation, Gibbs sampling, particle filters

1. INTRODUCTION

Modern Automatic Speech Recognition (ASR) systems work well when they are trained in an environment that matches well with the testing environment. However, when there is an acoustic mismatch between the training and the testing conditions, the performance is significantly inferior to what is achieved by a typical human listener. Many approaches have been adopted to overcome the degradation of ASR systems in adverse conditions. At the model level, hidden Markov models (HMM), which are a standard in modern ASR systems, can be adapted using maximum a priori (MAP) [1], maximum linear likelihood regression (MLLR) [2] or their variants. At the feature stage, vector Taylor series (VTS) [3], cepstral mean subtraction (CMS) [4] and ETSI [5] are some of the techniques that have been shown to work well and have been widely adopted in the speech recognition community.

Despite improvements achieved by these techniques, modern ASR systems under-perform significantly compared to a human listener who can achieve an intelligibility of $70 - 80\%$ at SNRs as low as $-6dB$ [6]. If the speech and noise sources are separated by an angle of 90° from one another, this intelligibility can be maintained even at $-16dB$. The superior performance in human perception can partly be attributed to their ability to track speech in the presence of other interfering signals [7].

Tracking the speech signal of interest can potentially improve the performance of ASR systems also. Nevertheless, conventional tracking techniques such as Kalman Filter [8] and extended Kalman filter [9] cannot be used for tracking the speech signal because state transition models are not available for speech and the distortion models in the presence of noise are highly non-linear in the feature domain. Monte Carlo methods are numerical methods based on random sampling and do not require analytical solutions to solve problems. Therefore, these methods can be deployed in situations where analytical solutions do not exist. Sequential Monte Carlo methods, also commonly known as particle filters, were first used in speech recognition paradigm in [10] [11] [12]. The particle filter in these algorithms did not track the speech signal directly, but instead tracked the noise signal, which was subsequently used for estimation of the clean speech signal. A more direct speech tracking approach was proposed in particle filter compensation (PFC) [13] [14], in which the clean speech distribution was estimated using an importance sampling scheme, where samples were generated using statistics from HMMs and the weights of these samples were computed using the distortion model of the speech signal under noisy conditions [13]. Further, a joint speech and noise tracking algorithm was proposed in [15], where noise is tracked using a particle filter that runs in parallel to the speech tracking algorithm.

In this paper, we propose a Markov chain Monte Carlo (MCMC) method for simultaneously estimating the distributions of the speech and noise components of noisy speech. The advantage of MCMC is that the noise is generated using multiple samples from the speech distribution and similarly, speech is updated using information from the noise samples. This ensures a better coupling between estimation of speech and noise when compared to the parallel tracking approaches [15], where a point estimate of speech is used in noise tracking.

The MCMC framework for joint clean speech and noise estimation is tested on the Aurora-2 connected digit recogni-

tion task. We achieve an error reduction of 12.87% at $0 - 20dB$ when compared with multi-condition trained models. The performance is better then the cases where only speech is tracked (error reduction improves slightly) and where speech and noise are tracked by two separate particle filters running in parallel.

2. PARTICLE FILTER APPROACH TO SPEECH FEATURE COMPENSATION

Speech tracking using PFC algorithm is summarized in the following steps [13]:

1. Posterior density of speech, based on the current observation is represented by a finite number of set points,

$$p(x_t|y_{0:\,t}) = \sum_{s=1}^{N_s} w_t^{(s)} \delta(x_t - x_t^{(s)}) \tag{1}$$

where $x_t^{(s)}$ for $s = 1, ..., N_s$ are the support points of PF.

2. The weight vector, $w_t^{(s)}$,associated with the support points[16] is computed with:

$$w_t^{(s)} = w_{t-1}^{(s)} \frac{p(y_t|x_t^{(s)}) p(x_t^{(s)}|x_{t-1}^{(s)})}{q(x_t^{(s)}|x_{t-1}^{(s)}, y_t)} \tag{2}$$

3. PFC is done in the spectral domain. Given additive noise with no channel effects [17], we have

$$y = x + \log(1 + e^{n-x}) \tag{3}$$

Then, we can evaluate $p(y|x)$ using

$$\begin{aligned} p(y|x) &= F'(u) \\ &= p(u) \frac{e^{y-x}}{e^{y-x} - 1} \end{aligned} \tag{4}$$

where x represents clean speech and n represents the noise.

4. The density $q(x_t^{(s)}|x_{t-1}^{(s)}, y_t)$ is used to generate the particle samples. In the PFC approach, we cluster the HMM states into M clusters and then use one of these clusters to generate the particle samples.

5. After generating samples, the weight for each sample is computed. We estimate the compensated features using discrete approximation of the expectation as

$$x_t = \sum_{s=1}^{N_s} w_t^{(s)} x_t^{(s)} \tag{5}$$

3. SPEECH AND NOISE ESTIMATION USING MCMC

For the joint estimation of the noise and the speech signal, the goal is to generate the posterior approximation for both signals and then make inferences from these approximate distributions. Consider a set of parameters $a_1, a_2, ..., a_p$ such that we want to estimate $p(a_1, a_2, ..., a_p|y_1, y_2, ..., y_n)$, where $y_1, y_2, ..., y_n$ is the set of observations. Given a starting point $a_1^{(0)}, a_2^{(0)}, ..., a_p^{(0)}$, the MCMC generates $a_j^{(s)}$ from $a_1^{(s-1)}, a_2^{(s-1)}, ..., a_{j-1}^{(s-1)}, a_{j+1}^{(s-1)}, ..., a_p^{(s-1)}$ as follows [18]

1. Sample $a_1^{(s)} \sim p(a_1^{(s)}|a_2^{(s-1)}, a_3^{(s-1)}, ..., a_p^{(s-1)})$
2. Sample $a_2^{(s)} \sim p(a_2^{(s)}|a_1^{(s-1)}, a_3^{(s-1)}, ..., a_p^{(s-1)})$

   $\vdots$
3. Sample $a_p^{(s)} \sim p(a_1^{(s)}|a_2^{(s-1)}, a_3^{(s-1)}, ..., a_{p-1}^{(s-1)})$

The resultant samples can be seen from two different perspectives. First, the sequence of the parameters $\mathbf{a^{(1)}}, \mathbf{a^{(2)}}, ..., \mathbf{a^{(s)}}$, where $\mathbf{a^{(s)}} = \{a_1^{(s)}, a_2^{(s)}, ..., a_p^{(s)}\}$, are dependent. The parameter $\mathbf{a^{(s)}}$ is conditionally independent on $\mathbf{a^{(1)}}, \mathbf{a^{(2)}}, ..., \mathbf{a^{(s-2)}}$ given $\mathbf{a^{(s-1)}}$ and therefore called the Markov chain. The other perspective is that the marginal distribution of parameter $\mathbf{a_j}$ is given by $a_j^{(0)}, a_j^{(1)}, ..., a_j^{(s)}$. This sampling distribution approaches the target distribution as $s \to \infty$. These samples can also be used to approximate the expected value of a function g using

$$\frac{1}{S} \sum_{s=1}^{S} g(\mathbf{a_j}) \to E[g(\mathbf{a_j})] \tag{6}$$

The MCMC approach allows us to estimate the joint distribution $p(n_t, x_t|y_t)$, from which we can extract the marginal distribution of interest $p(x_t|y_t)$. The straight forward implementation of MCMC described above requires that we can generate samples directly from

$$\begin{aligned} &p(n_t|x_t, y_t) \\ &p(x_t|n_t, y_t) \end{aligned} \tag{7}$$

However, it's not possible to generate samples from these distributions. To overcome this problem, the more general Metropolis algorithm can be used [19]. Specifically, we employ a combination of importance sampling and Metropolis sampling algorithm. Compared to the jumping distribution used in Metropolis-Hastings algorithm [20], where samples are rejected with probability r, the importance sampling scheme is more efficient. Another motivation for using importance sampling is the availability of the required framework from [13] [14]. We can evaluate $p(x_t^{(s)}|n_t^{(s-1)}, y_t)$ using

$$p(y_t|x_t^{(s)}) = p(u_t) \frac{e^{y_t - x_t^{(s)}}}{e^{y_t - x_t^{(s)}} - 1} \tag{8}$$

where $x_t^{(s)}$ represents the s^{th} clean speech sample at time t and the noise density is given by $\mathcal{N}(n_t^{(s-1)}, \sigma_n)$ and $u_t = \log(e^{y_t - x_t^{(s)}} - 1) + x_t$ with $F(u_t)$ being the Gaussian cumulative function with mean $n_t^{(s-1)}$ and variance σ_n^2. Similarly, to evaluate $p(n_t^{(s)}|x_t^{(s-1)}, y_t)$, we make use of the relation [15]

$$p(y_t|n_t) = \frac{1}{\sqrt{2\pi\sigma_{x,t}^2}} \exp[-\frac{1}{2\sigma_{x,t}^2}(y_t - \log(1 + \exp(n_t^{(s)} - x_t^{(s-1)})) - x_t^{(s-1)})^2] \quad (9)$$

The conditional distributions cover only a part of the distribution approximation by providing a mechanism to evaluate the weights of the samples. Prior to that, samples have to be available at the right locations. To generate the speech samples, we use the statistics available from HMMs. It is important to emphasize here that sample generation for x_t is not dependent on the n_t samples, rather, it is the computation of weights for x_t that is conditioned on noise samples. On the contrary, the location of samples and the weights for noise samples are both conditioned on the clean speech samples.

For each frame t, the algorithm proceeds as follows:

1. Generate sample x_t^s using HMMs
2. Compute weight for x_t^s using n_t^{s-1} in Eq. (8)
3. Generate sample n_t^s from x_t^{s-1} using Eq. (3)
4. Compute weight for n_t^s using x_t^{s-1} Eq. (5)
5. Repeat if $s < N_s$

where N_s is the desired number of samples. Once the point density of the clean speech features is available, we estimate the compensated features using Eq. (5).

4. COMPARISON OF PFC AND MCMC APPROACHES

The comparison of the PFC approach for speech compensation and the MCMC approach is laid out in Figure 1. The dashed arrows indicate the inter-dependencies between the speech distribution and the noise distribution. The cluster selection mechanism is the same for the PFC and the MCMC method. The speech samples, generated from the selected cluster, directly influence the generation of the noise samples, thereby ensuring a tight coupling between the speech and the noise samples. The noise statistics are updated using the approximation represented by the noise samples and their weights. Whereas the weights of the speech samples in the PFC algorithm are computed using noise statistics collected from the background, the speech samples weights in the MCMC algorithm are computed using noise statistics that are updated dynamically during the utterance. Since the speech distribution is approximated using noise statistics more specific to the current frame in MCMC, the approximation is improved compared to PFC.

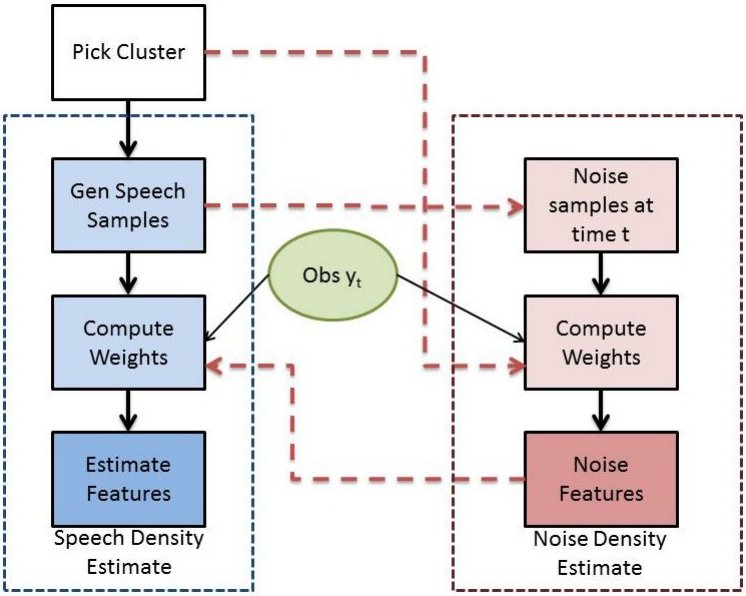

Fig. 1. Comparison of PFC and MCMC

5. EXPERIMENTS AND ANALYSIS

To evaluate the proposed framework, we experimented on the Aurora-2 connected digit recognition task. Compensation is carried out in the 23 channel fbank feature domain. From the test speech, we extracted the 39 element features (13 MFCCs and their first and second time derivatives) as well as 23 channel filter-bank features thereby forming two streams. The $1-best$ transcript, used for cluster selection, was determined by evaluating the MFCC stream with MC models. The speech samples are generated using the selected clusters. This helps us in preventing the sticking problem, which is a vulnerability of the the MCMC algorithms. If the speech sample is generated based on the noise sample and vice versa, then from equation (3), the samples would be concentrated in very small regions. This is so because after y_t is observed, knowing one of the n_t and x_t gives us quite precise information of the other. The sequence of samples generated for the speech and the noise signals is shown in Figure 2. The dependence of the noise samples is observable. The noise samples are related to the speech samples through the observation. Note that for higher value of the speech sample, the value for noise sample is smaller and vice versa.

The improvement obtained with MCMC in terms of error reduction in *word accuracy* over multi-condition training is given in Table 1. The second column of the table details the error reduction obtained with PFC over MC. The recognition performance for MCMC algorithm, when compared with PFC in terms of error reduction in *word accuracy*, improves for all noise levels except at lower SNRs i.e. $0dB$ and

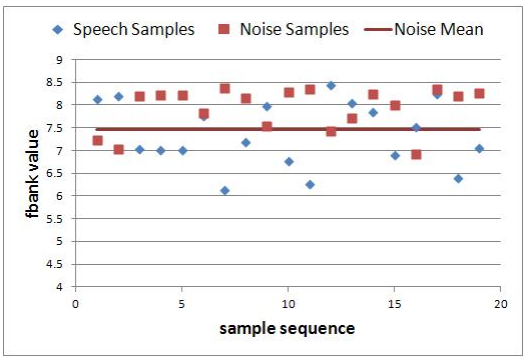

Fig. 2. Sample sequence for MCMC

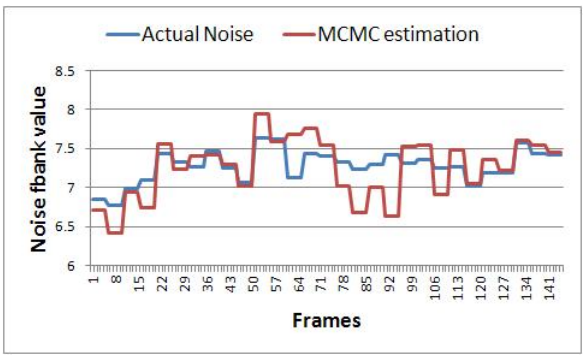

Fig. 3. Actual vs. estiamted noise at 10dB noise level

$-5dB$ (15.97% for PFC versus 15.26% for MCMC). Since the noise signal is dominant compared to the speech signal at these SNRs, the assumption that the noise samples can be well placed by indirectly using the speech statistics does not seem to hold. This results in poor noise estimates, and consequently, degradation in the recognition performance. Overall, an error reduction of 12.87% is achieved over multi-condition training compared to the 12.16% obtained with PFC for the $0dB - 20dB$ range.

Table 1. Error reduction over MC

ER	MCMC	PFC
20db	20.1%	14.5%
15db	20.3%	10.6%
10db	8.9%	1.9%
5db	8.1%	4.1%
0db	10.35%	12.63%
0-20db	12.87%	12.16%

The noise information from multiple frames, $t - N_p/2$ to $t + N_p/2$ (N_p frames), is combined together and used to recompute the weight of the x_t samples, i.e., the weight for the speech samples is recomputed after getting the noise estimate from N_p frames. Combining noise information from multiple frames improves the performance of the recognizer. Whereas the *word accuracy* for $N_p = 30$ is 90.2%, the corresponding performance for $N_p = 60$ and $N_p = 15$ is 90.1% and 90.06% respectively. The performance is inferior if N_p is either smaller or larger than the value 30. If N_p is large, the smaller variations in the noise estimate are averaged out and the performance is comparable to the case where the noise statistics are considered to be non-varying. On the other hand, the reduction in performance for N_p smaller than 30 is due to the erroneous estimate of noise in the smaller intervals. The noise estimation for a particular fbank channel is depicted in Figure 3. The errors in the noise estimate cause corresponding fluctuations in the compensated speech estimate. These fluctuations are undesirable from the machine learning perspective. For a better recognition performance, it is desirable that the behavior is consistent not only within training data but also between training and test data. The random variations caused by wrong noise estimate introduces discrepancies in both training and testing data. Increasing the number of frames, however, averages out these variations and improves the recognition performance.

The performance achieved with MCMC is also better than the improvement obtained from tracking the speech signal using a particle filter that is running parallel to the PFC algorithm [15]. The comparison for subway noise is given in Table 2. Both approaches still fall short of the case where exact noise information (average over 30 frames) is available and therefore, the margin for improvement in noise estimation is still present.

Table 2. Noise Estimation Comparison

Word Accuracy	Joint Noise Tracking With MCMC	Joint Noise Tracking With Two PFs	Noise Known
0-20dB	91.25%	91.18%	91.41%

6. CONCLUSIONS

We have proposed a joint clean speech and noise tracking approach using an MCMC algorithm. The noise samples are dependent on the location of the speech samples which are generated using the statistical information available from the HMMs. Results show that estimating noise information by MCMC algorithm and then using this information for speech compensation improves the speech recognition performance. However, a margin of improvement still exists to come at par with the case when noise information is better known. In future, other MCMC techniques will be explored to improve the noise estimate and thus the ASR performance.

7. REFERENCES

[1] J.L. Gauvain and C.-H. Lee, "Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains," *Speech and Audio Processing, IEEE Transactions on*, vol. 2, no. 2, pp. 291–298, 1994.

[2] CJ Leggetter and PC Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models," *Computer speech and language*, vol. 9, no. 2, pp. 171, 1995.

[3] A. Acero, L. Deng, T. Kristjansson, and J. Zhang, "Hmm adaptation using vector taylor series for noisy speech recognition," in *Sixth International Conference on Spoken Language Processing*, 2000.

[4] H.K. Kim and R.C. Rose, "Cepstrum-domain acoustic feature compensation based on decomposition of speech and noise for asr in noisy environments," *Speech and Audio Processing, IEEE Transactions on*, vol. 11, no. 5, pp. 435–446, 2003.

[5] D. Macho, L. Mauuary, B. Noé, Y.M. Cheng, D. Ealey, D. Jouvet, H. Kelleher, D. Pearce, and F. Saadoun, "Evaluation of a noise-robust dsr front-end on aurora databases," in *Seventh International Conference on Spoken Language Processing*, 2002.

[6] P. Assmann and Q. Summerfield, "The perception of speech under adverse conditions," *Speech processing in the auditory system*, pp. 231–308, 2004.

[7] S. Greenberg and W. Ainsworth, *Listening to speech: an auditory perspective*, Lawrence Erlbaum, 2006.

[8] R.G. Brown and P.Y.C. Hwang, "Introduction to random signals and applied kalman filtering. 1997," *NY John Wiley and Sons*.

[9] S. Haykin, *Adaptive Filter Theory, 2002*, Prentice-Hall.

[10] R. Singh and B. Raj, "Tracking noise via dynamical systems with a continuum of states," in *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*. IEEE, 2003, vol. 1, pp. I–396.

[11] B. Raj, R. Singh, and R. Stern, "On tracking noise with linear dynamical system models," in *Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on*. IEEE, 2004, vol. 1, pp. I–965.

[12] M. Fujimoto and S. Nakamura, "Sequential non-stationary noise tracking using particle filtering with switching dynamical system," in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*. IEEE, 2006, vol. 1, pp. I–I.

[13] A. Mushtaq, Y. Tsao, and C.-H. Lee, "A particle filter feature compensation approach to robust speech recognition," in *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

[14] A. Mushtaq and C.-H. Lee, "An integrated approach to feature compensation combining particle filters and hidden markov model for robust speech recognition," in *Acoustics, Speech and Signal Processing, 2012. ICASSP 2012 Proceedings. 2012 IEEE International Conference on*. IEEE, 2012, vol. 1, pp. I–I.

[15] A. Mushtaq and C.-H. Lee, "Joint tracking of clean speech and noise using hmms and particle filters for robust speech recognition," *Asilomar Conference on Signals, Systems and Computers*, 2012.

[16] M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *Signal Processing, IEEE Transactions on*, vol. 50, no. 2, pp. 174–188, 2002.

[17] P.J. Moreno, B. Raj, and R.M. Stern, "A vector taylor series approach for environment-independent speech recognition," in *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*. IEEE, 1996, vol. 2, pp. 733–736.

[18] P.D. Hoff, *A first course in Bayesian statistical methods*, Springer, 2009.

[19] A.E. Gelfand and A.F.M. Smith, "Sampling-based approaches to calculating marginal densities," *Journal of the American statistical association*, vol. 85, no. 410, pp. 398–409, 1990.

[20] S. Chib and E. Greenberg, "Understanding the metropolis-hastings algorithm," *The American Statistician*, vol. 49, no. 4, pp. 327–335, 1995.