A Particle Filter Feature Compensation Approach to Robust Speech Recognition

Aleem Mushtaq[1], *Yu Tsao*[2] and *Chin Hui-Lee*[1]

[1]School of ECE, Georgia Institute of Technology, Atlanta, GA, 30332-0250, USA
[2]National Institute of Information and Communications Technology, Kyoto, Japan
aleem@gatech.edu, yu.tsao@nict.go.jp, chl@ece.gatech.edu

Abstract

We propose a novel particle filter approach to enhancing speech features for robust speech recognition. We use particle filters to compensate the corrupted features according to an additive noise distortion model by incorporating both the statistics from the clean speech Hidden Markov Models and of the observed background noise to map the noisy features back to clean speech features. We report on experimental results obtained with the Aurora-2 connected digit recognition task, and show that a large digit error reduction of 67% from multi-condition training is attainable if the missing side information needed for particle filter based compensation were available. When such nuisance parameters are estimated in actual operational conditions then an error reduction of only 13% is currently achievable. We anticipate more improvements in the future when better estimation algorithms are explored.

Index Terms: robustness, noise compensation, particle filter

1. Introduction

Most state-of-the-art automatic speech recognition (ASR) algorithms employ hidden Markov models (HMMs) to characterize and decode speech utterances. However the performance of ASR systems often degrades in adverse conditions when there is a potential acoustic mismatch between training and testing conditions. One way to alleviate this difficulty is to start with a model which is environment independent and then adapt it to a specific testing condition using a small amount of data. Another way is to compensate features to the test condition of interest using data from that environment and subsequently applying this transformation to the environment independent models.

To compensate speech distortion, the feature vector is often approximated with a vector Taylor series (VTS) expansion [1-2]. Schemes, such as cepstral mean subtraction [3] and ETSI advanced front-end (AFE) [4], have also been adopted. Recently, particle filter based methods have been attempted for the case with additive noise [6-8]. A sequential Monte Carlo feature compensation algorithm was initially proposed [6-7] in which the noise was treated as a state variable while speech was considered as the signal corrupting the observation noise and a VTS approximation was used to approximate the clean speech signal by applying a minimum mean square error (MMSE) procedure. In [8] extended Kalman filters were used to model a dynamical system representing the noise which was further improved by using Polyak averaging and feedback with a switching dynamical system [9]. The previous attempts to incorporate particle filter for speech recognition have been more indirect as it was used for tracking of noise instead of the speech signal itself. Since the speech signal is treated as corrupting signal to the noise, limited or no information readily available from the HMMs or the recognition process can be utilized efficiently in the compensation process.

Particle filters are powerful numerical mechanisms for sequential signal modeling and is not constrained by the conventional linearity and Gaussianity [5] requirements. It is a generalization of the Kalman filter [10] and is more flexible than the extended Kalman filter [11] because the stage-by-stage linearization of the state space model in Kalman filter is no longer required [5]. One difficulty of using particle filters lies in obtaining a state space model for speech as consecutive speech features are usually highly correlated. Just like in the Kalman filter and HMM frameworks, state transition is an integral part of the particle filter algorithms.

In contrast to the previous particle filter attempts [6-8] we treat the speech signal as the state variable and the noise as the corrupting signal and attempt to estimate clean speech from noisy speech. We incorporate statistical information available in the acoustic models of clean speech, e.g., the HMMs trained with clean speech, as an alternative state transition model. The similarity between HMMs and particles filters can be seen from the fact that an observation probability density function corresponding to each state of an HMM describes, in statistical terms, the characteristics of the source generating a signal of interest if the source is in that particular state, whereas in particle filters we try to estimate the probability distribution of the state the system is in when it generates the observed signal of interest. Particle filters are suited for feature compensation because the probability density of the state can be updated dynamically on a sample-by-sample basis. On the other hand, state densities of the HMMs are assumed independent of each other. Although they are good for speech inference problems, HMMs do not adapt well in fast changing environments.

By establishing a close interaction of the particle filters and HMMs, the potentials of both models can be harnessed in a joint framework to perform feature compensation for robust speech recognition. Just like in iterative maximum likelihood *stochastic matching* [12] we improve the recognition accuracy through compensation of noisy speech, and we enhance the compensation process by utilizing information in the HMM state transition and mixture component sequences obtained in the recognition process. This can be seen in Figure 1.

When state sequence information is available we found we can attain a 67% digit error reduction from multi-condition training in the Aurora-2 connected digit recognition task. If the missing parameters are estimated in the operational situations we only observe a 13% error reduction in the current study. Moreover, by tracking the speech features, compensation can be done using only partial information about noise and consequently good recognition performance can be obtained despite potential distortion caused by non-stationary noise within an utterance. More powerful estimation algorithms will therefore be explored in the future.

2. Background and Motivation

The mismatch between training and testing data can be viewed in the signal, feature and model space [12]. Using particle filter algorithms with side information about the statistics of clean speech available in the clean HMMs we can perform feature compensation. If the clean speech is corrupted by an additive noise, n, and a distortion channel, h, then we can represent the noise corrupted speech with an additive noise model [2], assuming known statistics of the noise parameters,

$$y = x + h + \log(1 + \exp(n - x - h)) \quad (1)$$

where $y = \log(S_y(m_p))$, $x = \log(S_x(m_p))$, $h = \log(|H(m_p)|^2)$ and $S_y(m_p) = S_x(m_p)|H(m_p)|^2 + S_N(m_p)$ and $S(m_p)$ denotes the p^{th} mel spectrum. The additional side information needed for feature compensation is a set of nuisance parameters, Φ. Similar to *stochastic matching* [12], we can iteratively find Φ followed by decoding as shown in Figure 1:

$$\Phi' = \arg\max_{\Phi} P(Y' | \Phi, \Lambda) \quad (2)$$

where Y' is the noisy or compensated utterance.

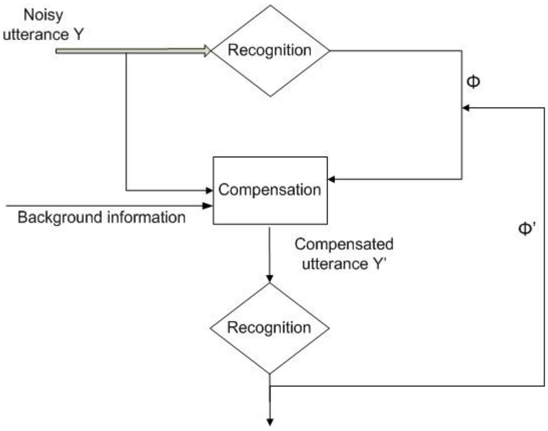

Figure 1. A general feature compensation scheme

3. Particle Filter Based Compensation

Particle filtering is a way to model signals emanating from a dynamical system. If the underlying state transition is known and the relationship between the system state and the observed output is available, then the system state can be found using Monte Carlo simulations [13]. In this paper we propose using particle filter to estimate noise-compensated features. Consider the discrete time Markov process such that

$$\begin{aligned} &X_1 \sim \mu(x_1) \\ &X_t | X_{t-1} = x_t \sim p(x_t | x_{t-1}) \\ &Y_t | X_t = x_t \sim p(y_t | x_t) \end{aligned} \quad (3)$$

We are interested in obtaining $p(x_t | y_{1:t})$ so that we have a filtered estimate of x_t from the measurements available so far, $y_{1:t}$. If the state space model for the process is available, and both the state and the observation equations are linear, then Kalman filter [10] can be used to determine the optimal estimate of x_t given observations $y_{1:t}$. This is so under the condition that process and observation noises are white Gaussian noise with zero mean and mutually independent. In case the state and observation equations are nonlinear, the Extended Kalman Filter (EKF) [11], albeit suboptimal, can be used. Alternatively, if the state space can be represented by a finite number of states as in the case of modeling a speech utterance then it is natural to consider using HMMs to find the state sequence that best describes the observed data sequence $y_{1:t}$ in a probabilistic sense. Particle filter compensation (PFC) estimates the features with the posterior density, $p(x_t | y_{1:t})$, represented by a finite set of support points [5]:

$$p(x_{0:t} | y_{1:t}) = \sum_{i=1}^{N_s} w_t^i \delta(x_{0:t} - x_{0:t}^i) \quad (4)$$

where $x_{0:t}^i$ for $i = 0, \ldots, N_s$ are the support points and w_t^i are the associated weights. We thus have a discretized and weighted approximation of the posterior density without the need of an analytical solution. The support points are determined based on the concept of importance sampling in which instead of drawing from $p(.)$, we draw points from another distribution $q(.)$ and compute the weights using the following:

$$w^i = \frac{\pi(x^i)}{q(x^i)} \quad (5)$$

where $\pi(.)$ is the distribution of $p(.)$ and $q(.)$ is an importance density from which we can draw samples. For the sequential case, the weight update equation can be computed one by one,

$$w_t^i = w_{t-1}^i \frac{p(y_t | x_t^i) p(x_t^i | x_{t-1}^i)}{q(x_t^i | x_{t-1}^i, y_t)} \quad (6)$$

and Eq. (4) can be rewritten as

$$p(x_t | y_{o:t}) = \sum_{i=1}^{N_s} w_t^i \delta(x_t - x_t^i) \quad (7)$$

We still need a state transition model to derive $q(.)$ before we can implement the PFC algorithm. Since HMMs are used to characterize speech utterances we propose using HMM transitions to generate samples from $q(x_t | x_{t-1}^i, y_t)$.

4. PFC Algorithm Implementation

The clean HMMs and the background noise information enable us to generate appropriate samples from $q(.)$ in Eq. (6). The parameters, Φ, in Eq. (2) in our PFC implementation, correspond to the corresponding correct HMM state sequence and mixture component sequence. These sequences provide critical information for density approximation in PFC. As shown in Figure 1 this can be done in two stages. We first perform a front-end compensation of noisy speech. Then recognition is done in the second stage to generate the side information, Φ, so as to improve compensation. This process can be iterated similar to what's done in maximum likelihood stochastic matching [12]. During compensation, the observed speech y is mapped to clean speech features, x. For this purpose clean speech alone cannot be represented by a finite set of points and therefore HMMs by themselves cannot be used directly for tracking of x. Now if an HMM λ_m is available that adequately represents the speech segment under consideration for compensation with an estimated state sequence $s_1, s_2, \ldots, s_T$ that correspond to T feature vectors to be considered in the segment, then we can generate the samples from the i^{th} sample according to

$$p(x_t | x_{t-1}^i) \sim \sum_{k=1}^{K} c_{k,s_t} N(\mu_{k,s_t}, \Sigma_{k,s_t}) \quad (8)$$

where $N(\mu_{k,s_t}, \Sigma_{k,s_t})$ is the k^{th} Gaussian mixture for the state s_t in λ_m and c_{k,s_t} is its corresponding weight for the mixture. The total number of particles is fixed and the contribution from each mixture, computed at run time, depends on its weight. We have chosen the *importance sampling density,* $q(x_t | x_{t-1}^i, y_t)$, in Eq. (6) as $p(x_t | x_{t-1}^i)$ in Eq. (8). This is known as the sampling importance resampling (SIR) filter [5]. It is one of the simplest implementation of particle filters and it enables the generation of samples independently from the observation. For the SIR filter, we only need to know the state and the observation equations and should be able to sample from the prior as in Eq. (3). Also, the resampling step is applied at every stage and the weight assigned to the i-th support point of the distribution of the speech signal at time t is updated as:

$$w_t^i \propto p(y_t \mid x_t^i) \tag{9}$$

The procedure for obtaining HMMs and the state sequence will be described in detail later. To obtain $p(y_t \mid x_t^i)$, the distribution of the log spectra of noise for each channel is assumed Gaussian with mean μ_n and variance σ_n^2. Assuming there is additive noise only with no channel effects

$$y = x + \log(1 + e^{n-x}) \tag{10}$$

We are interested in evaluating $p(y \mid x)$ where x represents clean speech and n is the noise with density $N(\mu_n, \sigma_n)$. Then

$$\begin{aligned} p[Y < y \mid x] &= p[x + \log(1 + e^{N-x}) < y \mid x] \\ p(y \mid x) &= F'(u) = p(u)\frac{e^{y-x}}{e^{y-x} - 1} \end{aligned} \tag{11}$$

Where $F(u)$ is the Gaussian cumulative density function with mean μ_n and variance σ_n^2 and $u = \log(e^{y-x} - 1) + x$. In the case of MFCC features, the nonlinear transformation is [1]

$$y = x + D\log(1 + e^{D^{-1}(n-x)}) \tag{12}$$

Consequently,

$$p(y \mid x) = p_N(g^{-1}(y))J_{g^{-1}}(y) \tag{13}$$

where $p_N(.)$ is a Gaussian pdf, $J_{g^{-1}}(y)$ is the corresponding Jacobian and D is a discrete cosine transform matrix which is not square and thus not invertible. To overcome this problem, we zero-pad the y and x vectors and extend D to be a square matrix. The variance of the noise density is obtained from the available noise samples. Once the point density of the clean speech features is available, we estimate of the compensated features using discrete approximation of the expectation as

$$x_t = \sum_{i=1}^{N_S} w_t^i x_t^i \tag{14}$$

where N_s is the total number of particle samples at time t.

5. Estimation of HMM Side Information

As described above, it is important to obtain $\Phi \in \{\lambda_m, S\}$ where λ_m is an HMM that faithfully represents the speech segment being compensated and $S = s_1, s_2, ..., s_T$ is the state sequence corresponding to the utterance of length T. To obtain λ_m for the m^{th} word W_m in the utterance, we chose the N-best models $\lambda_{m_1}, \lambda_{m_2}, ..., \lambda_{m_N}$ from HMMs trained using 'clean speech data'. The N models are combined together to obtain a single model λ_m as follows.

5.1 Gaussian Mixtures Estimation

To obtain the observation model for each state j of model λ_m, we concatenate mixtures from the corresponding states of all component models,

$$\hat{b}_j^{(m)}(o) = \sum_{l=1}^{L}\sum_{k=1}^{K} c_{k,j}^{(m_l)} N(\mu_{k,j}^{(m_l)}, \Sigma_{k,j}^{(m_l)}) \tag{15}$$

where K is the number of Gaussian mixtures in each original HMM and L is the number of different words $m_1, m_2, ..., m_l$ in the N-best hypothesis. $\mu_{k,j}^{(m_l)}$ and $\Sigma_{k,j}^{(m_l)}$ are mean and covariance from the k^{th} mixture in the j^{th} state of model m_l. The mixture weights are normalized by scaling them according to the likelihood of the occurrence of the model, from which they come from,

$$c_{k,j}^{(m_l)} = c_{k,j}^{(m_l)} \times p(W_m = \lambda_{m_l}) \tag{16}$$

The mixture weight is an important parameter because it determines the number of samples that will be generated from the corresponding mixture. The state transition coefficients for λ_m are computed using the following:

$$\begin{aligned} \hat{a}_{ij}^{(m)} &= \sum_{l=1}^{L} p[s_t^{(m_l)} = i, s_{t-1}^{(m_l)} = j \mid W_m = \lambda_{m_l}]p[W_m = \lambda_{m_l}] \\ \hat{a}_{ij}^{(m)} &= \sum_{l=1}^{L} [a_{ij}^{(m_l)} \mid W_m = \lambda_{m_l}]p[W_m = \lambda_{m_l}] \end{aligned} \tag{17}$$

5.2 State Sequence Estimation

The recognition performance can be greatly improved if a good estimate of the HMM state sequence S is available. But obtaining this sequence in a noisy operational environment in ASR is very challenging. The simplest approach is to use the decoded state sequence obtained with multi-condition trained models in an ASR recognition process as shown in the bottom of Figure 1. However, these states could often correspond to incorrect models and deviate significantly from the optimal one. Alternatively, we can determine the states (to generate samples from) sequentially during compensation. For left-to-right HMMs, given the state s_{t-1} at time $(t-1)$, we chose s_t using (18) as follows:

$$\begin{aligned} s_t &\sim a_{s_t, s_{t-1}} \\ s_t &= \arg\max_i(a_{ij}) \end{aligned} \tag{18}$$

where a comes from the state transition matrix for λ_m and $i = j$ or $i = j+1$. The mixture indices are subsequently selected from amongst the mixtures corresponding to the chosen state. We can generate samples from all mixtures or we can ignore mixtures whose weight is below a certain threshold.

6. Experimental Results and Analysis

To investigate the properties of the proposed approach, we first assume that a decent estimate of the state is available at each frame. Moreover, we assume that speech boundaries are marked and therefore the silence and speech sections of the utterance are known. To obtain this information, we use a set of digit HMMs (18 states, 3 Gaussian mixtures) that have been trained using clean speech represented by 23 channel mel-scale log spectral feature. The speech boundaries and state information for a particular noisy utterance is then captured through digit recognition performed on the corresponding clean speech utterance. The speech boundary information is critical because the noise statistics have to be estimated from the noisy section of the utterance. To get the HMM needed for particle filter compensation L models $\lambda_1, \lambda_2, \ldots, \lambda_L$ are selected based on the N-best hypothesis list. For our experiments, we set $L = 3$. We combine these models to get λ_m' for the m^{th} word in the utterance. Best results are obtained if the correct word model is present in the pool of models that contribute to λ_m'. Upon availability of this information, the compensation of the noisy log spectral features is done using the sequential importance sampling. To see the efficacy of the compensation process, we consider the noisy, clean and compensated filter banks (channel 8) for the whole utterances shown in Figure 2. The SNR for this particular case is 5 dB. When compared with the noisy feature we can see that the compensated feature matches well with the clean feature. It should be noted however that such a good restoration of the clean speech signal from the noisy signal is achievable only when a good estimate of the side information about the state and mixture component sequences is available.

Assuming all such information were given (the ideal oracle case) recognition can be performed on MFCCs (39 MFCCs with 13 MFCCs and their first and second time derivatives) extracted from these compensated log spectral features. The HMMs used for recognition are trained with noisy data that has been compensated in the same way as the testing data. The performance compared to multi-condition (MC) and clean condition training (Columns 5 and 6 in Table 1) is given in Column 2 of Table 1 (Adapted Model I). It is clearly noted that a very significant 67% digit error reduction was attained if the missing information were made available to us.

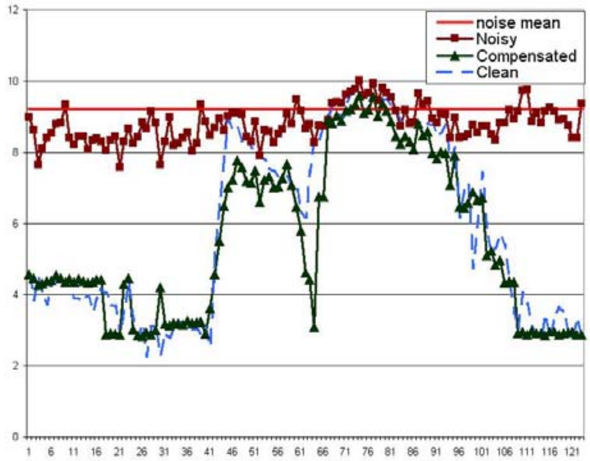

Figure 2. Fbank channel 8 corresponding to noisy (SNR = 5 dB) and compensated speech.

Table 1. ASR accuracy comparisons for Aurora-2

Word Accuracy	Adapted Models I	Adapted Models II	Adapted Models III	MC Training	Clean Training
clean	99.10	99.10	99.10	98.50	99.11
20dB	97.75	96.46	97.38	97.66	97.21
15dB	97.61	95.98	96.47	96.95	92.36
10dB	96.66	94.00	94.40	95.16	75.14
5dB	95.20	90.64	88.02	89.14	42.42
0dB	92.13	82.62	68.28	64.75	22.57
-5dB	89.28	72.13	32.92	27.47	NA
0-20dB	95.86	90.23	88.91	88.73	65.94

In the case of the actual operational scenarios, when no side information is available, models were chosen from the *N*-Best list while the states were computed using Viterbi decoding. Of course, the states would correspond to only one model which might not be correct, and there might be a significant mismatch between actual and computed states. Moreover the misalignment of words also exacerbated the problem. The results for this case (Adapted Model III as shown in Table 1 Column 4) were only marginally better than those obtained with the multi-condition trained models. To see the effects of the improvements for the case where the states are better aligned, we made use of whatever information we could get. The boundaries of words were extracted from the *N*-Best list using exhaustive search and the states for the words between these boundaries were assigned by splitting the digits into equal-sized segments and assigning one state to each segment. This limited the damage done by state misalignment, and it can be seen that a 13% digit error reduction from MC training was observed (Adapted Model II in Table 1 Column 3).

7. Summary

We propose a particle filter compensation approach to robust speech recognition, and show that a tight coupling and sharing of information between HMMs and particle filters has a strong potential to improve recognition performance in adverse environments. It is noted that we need an accurate alignment of the state and mixture sequences used for compensation with particle filters and the actual HMM state sequences that describes the underlying clean speech features. Although we have observed an improved performance in the current particle filter compensation implementation there is still a considerable performance gap between the oracle setup with correct side information and what's achievable in this study with the missing side information estimated from noisy speech.

We anticipate that the current performance gap can be narrowed when more advanced algorithms are explored to obtain better estimates of the missing side information needed to fully utilize the power of particle filter compensation.

8. Acknowledgments

The first author is supported in part by the Higher Education Commission of Pakistan Split PhD scholarship Program.

9. Reference

[1] A. Acero, L. Deng, T. Kristjansson, and J. Zhang, "HMM adaptation using vector Taylor series for noisy speech recognition," *Proc. ICSLP*, pp. 869-872, 2002.

[2] P. J. Moreno, "Speech recognition in noisy environments," Ph. D. Thesis, Department of ECE, Carnegie Mellon Univ., 1996.

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

[4] D. Macho, *et a*l, "Evaluation of a noise-robust DSR front-end on Aurora databases," *Proc. ICSLP*, 2002.

[5] M .S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking," *IEEE Trans. Signal Proc.*, 2002.

[6] K. Yao and S. Nakamura: "Sequential noise compensation by sequential Monte Carlo method," *Proc. NIPS*, 2001.

[7] B. Raj, R. Singh, and R. Stern, "On tracking noise with linear dynamical system models." *Proc. ICASSP*, 2004.

[8] M. Fujimoto and S. Nakamura, "Particle Filter based non-stationary noise tracking for robust speech recognition," *Proc. ICASSP*, 2005.

[9] M. Fujimoto and S. Nakamura, "Sequential non-stationary noise tracking using particle filtering with switching dynamical system," *Proc. ICASSP*, 2006.

[10] Robert Grover Brown and Patrick Y. C. Hwang. 1996. *Introduction to Random Signals and Applied Kalman Filtering*, *3rd edition*, Prentice Hall.

[11] Simon Haykin. 2009. *Adaptive Filter Theory*, 4th *edition*, Prentice Hall.

[12] A. Sankar and C.-H. Lee, "A maximum-likelihood approach to stochastic matching for robust speech recognition," *IEEE Trans. Speech Audio Processing*, vol. 4, pp.190-202, May.1996.

[13] N Arnaud, Doucet, and Johansen, "A tutorial on particle filtering and smoothing: Fifteen years later," Tech. Rep., 2008. [Online]. http://www.cs.ubc.ca/~arnaud/doucet_johansen_tutorialPF.pdf