AN INTEGRATED APPROACH TO FEATURE COMPENSATION COMBINING PARTICLE FILTERS AND HIDDEN MARKOV MODELS FOR ROBUST SPEECH RECOGNITION

Aleem Mushtaq and *Chin Hui-Lee*

School of ECE, Georgia Institute of Technology, Atlanta, GA, 30332-0250, USA
aleem@gatech.edu, chl@ece.gatech.edu

ABSTRACT

Obtaining accurate hidden Markov model (HMM) state sequences is a research challenge to warrant good system performance in particle filter (PF) compensation for noisy speech recognition. Instead of using specific knowledge at the model and state levels which is hard to estimate, we pool model states into clusters as side information. Since each cluster encompasses more statistics when compared to the original HMM states, there is a higher possibility that the newly formed probability density function at the cluster level can cover the underlying speech variation to generate appropriate PF samples for feature compensation. Testing the proposed PF-based compensation scheme on the Aurora 2 connected digit recognition task, we achieve an error reduction of 12.15% from the best multi-condition trained models using this integrated PF-HMM framework to estimate the cluster-based HMM state sequence information.

Index Terms— particle filter compensation, hidden Markov model, clustering, robust speech recognition

1. INTRODUCTION

State-of-the-art automatic speech recognition (ASR) algorithms use hidden Markov models (HMMs) for modeling speech and they often work very well in matched conditions. However, the performance degrades when there is a mismatch between training and testing conditions. To alleviate this problem, we can adapt the models to new conditions using techniques such as maximum a posteriori (MAP) [1], and maximum likelihood linear regression (MLLR) [2], etc. On the other hand, we can compensate the distorted speech features and attempt to map them to the space of speech features that were used to train the HMMs. Vector Taylor series (VTS) [3], cepstral mean subtraction (CMS) [4], and ETSI advanced front-end (AFE) [5] are notable examples of such approaches. Particle filter (PF) [6] is a numerical method to sequentially simulate a target distribution based on Monte Carlo sampling. PF-based compensation (PFC) proposed in [7] attempts to enhance speech features in order to improve noisy speech recognition. The clean speech model set is first tracked using particle filter algorithm in the filter bank domain and then ASR is performed on the mel-frequency cepstral coefficient (MFCC) features extracted from the newly estimated filter bank features. Compared to other PF based compensation techniques [8-9] PFC is a more direct approach to track clean speech features in the noisy environment. A direct approach enables us to obtain probability density of underlying speech dynamically on sample-by-sample basis. If this probability density is constructed accurately, considerable improvements in recognition results can be obtained. Moreover, compensation can be done using limited noise information, making the algorithm less susceptible to noise variations within an utterance.

Particle Filters are also superior to other tracking techniques such as Kalman filter [10] and extended Kalman filter [11] because it is not constrained by the conventional linearity and Gaussianity [6] requirements. However, particle filters do require a state space model, which is difficult to obtain for speech signals in the spectral domain. State transition information is an integral part of the particle filter algorithm and is used to propagate the particle samples through time transitions of the signal being processed. Specifically, the state transition is important to be able to position the samples at the right locations.

To solve this problem, statistics from HMMs can be used. Although we only have discrete states in HMMs, each state is characterized by a continuous density Gaussian mixture model (GMM) and therefore it enables us to capture part of the variation in speech features to generate particle samples for feature compensation. This setting is referred to as an integrated PF-HMM framework. A key problem to resolve here is the larger the number of states and the mixture Gaussian components in the HMM set, the harder it is to choose the correct models to generate particle samples for PF-based compensation. In this paper we propose a HMM state clustering approach to estimating the HMM cluster instead of state sequences. When dealing with non-stationary noise this allows us to dynamically track and compensate noisy speech features.

The integrated PF-HMM framework is tested on the Aurora 2 connected digit recognition task. It was found that the proposed PFC framework reduces the average digit error rates by 12.15% from 11.27%, obtained with the best multi-condition trained models, to 9.9%, for conditions with signal-to-noise ratios ranged from 0dB to 20dB, when the number of clusters and particles are chosen appropriately to estimate the HMM state information.

2. COMBINING PARTICLE FILTER AND HMM FOR SPEECH FEATURE COMPENSATION

Particle filtering is often used 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, the state can be found using Monte Carlo simulations [12]. Considering the underlying process to be discretely Markov:

$$X_1 \sim \mu(x_1)$$
$$X_t \mid X_{t-1} = x_t \sim p(x_t \mid x_{t-1}) \quad (1)$$
$$Y_t \mid X_t = x_t \sim p(y_t \mid x_t)$$

We estimate $p(x_t|y_{1:t})$ so that we have a filtered estimate of x_t from the measurements available so far, $y_{1:t}$. The particle filter compensation (PFC) process is summarized in 6 steps as follows:

1) Posterior density, $p(x_t|y_{1:t})$ is represented by a finite number of support points, x_t^i for $i = 0, \dots, N_S$ [6],
$$p(x_t \mid y_{o:t}) = \sum_{i=1}^{N_S} w_t^i \delta(x_t - x_t^i) \quad (2)$$
2) The weight vector, w^i, associated with the support points, approximates the posterior density and are determined based on the concept of *importance sampling* [6] computed with:
$$w_t^i = w_{t-1}^i \frac{p(y_t \mid x_t^i) p(x_t^i \mid x_{t-1}^i)}{q(x_t^i \mid x_{t-1}^i, y_t)} \quad (3)$$
3) PFC is done in spectral domain. This is so because fairly accurate models that describe the relationship between clean speech (the signal being tracked) and the noisy speech (the signal being observed) are available in the filter bank domain. Given additive noise with no channel effects [3],
$$y = x + \log(1 + e^{n-x}) \quad (4)$$
then we can evaluate $p(y \mid x)$ using
$$p(y \mid x) = F'(u) = p(u) \frac{e^{y-x}}{e^{y-x} - 1} \quad (5)$$
where x represents clean speech and n is the noise with density $N(\mu_n, \sigma_n)$, assuming each channel is Gaussian with mean μ_n and variance σ_n^2. Denoting $u = \log(e^{y-x} - 1) + x$, $F(u)$ is the Gaussian cumulative density function with mean μ_n and variance σ_n^2. The variance of the noise density can be obtained from the available noise-only frames.
4) The density $q(x_t|x_{t-1}^i, y_t)$ plays a crucial role in particle filtering known as the *importance sampling density* that is used to generate the particle samples. In case of speech signals, it is difficult to obtain because it is derived from the state transition model which is not available. If $q(.)$ can be constructed, weights and consequently the posterior density $p(x_t|y_{1:t})$ can then be easily evaluated. One main issue to be resolved is to find a suitable $q(.)$ based on the available information. In PFC, we use
$$q(x_t \mid x_{t-1}^i, y_t) \sim \sum_{k=1}^{K} c_{k,s_t} N(\mu_{k,s_t}, \Sigma_{k,s_t}) \quad (6)$$
to generate the samples, where $N(\mu_{k,s_t}, \Sigma_{k,s_t})$ is the k^{th} Gaussian mixture component for state s_t in model λ_m with c_{k,s_t} being its corresponding mixture weight. The specific model λ_m and a state sequence estimate, $s_1, s_2, \dots, s_T$ that adequately represents the speech segment can be obtained through recognition decoding using the multi-condition trained models.
5) After generating samples from Eq. (6), the weights are computed using Eq. (3). Once the point density of the clean speech features is available, we estimate the compensated features using discrete approximation of the expectation of the particle filter as
$$x_t = \sum_{i=1}^{N_S} w_t^i x_t^i \quad (7)$$
6) As mentioned above, compensated features are obtained in filter bank domain. To exploit the superior discrimination power of cepstrum, we transform the compensated filter bank features to MFCC. The final recognition result for the test utterance is then decoded from the compensated MFCC features.

3. A CLUSTERING APPROACH TO OBTAINING CORRECT HMM INFORMATION

It has been shown that choosing the correct HMM model and state sequences greatly improves the accuracy of the overall recognition system [7]. However in operational scenarios, it is often difficult to obtain such information using the multi-condition trained models. Moreover, if this information is far from ideal, the recognition performance could be worse than the baseline. To alleviate this problem, we adopt a clustering approach to simplify the process of picking the best distributions to generate the particle samples from.

As can be seen from Eq. (6), HMM states are used to spread the particles at the right locations for subsequent estimation of the underlying clean speech density. If the state is incorrect, the location of particles will be wrong and the density estimate will be erroneous. One solution is to merge the states into clusters. Since the total number of clusters can be much less than the number of states, the problem of choosing the correct information block for sample generation is simplified. A tree structure to group the Gaussian mixtures from clean speech HMMs into clusters can be built with the following distance measure [13]:
$$d(m,n) = \int g_m(x) \log \frac{g_m(x)}{g_n(x)} dx + \int g_n(x) \log \frac{g_n(x)}{g_m(x)} dx \quad (8)$$
$$= \sum_i [\frac{\sigma_m^2(i) - \sigma_n^2(i) + (\mu_n(i) - \mu_m(i))^2}{\sigma_n^2(i)} + \frac{\sigma_n^2(i) - \sigma_m^2(i) + (\mu_n(i) - \mu_m(i))^2}{\sigma_m^2(i)}] \quad (9)$$

where $\mu_m(i)$ is the i^{th} element of the mean vector $\mu(m)$ and $\sigma_m^2(i)$ is the i^{th} diagonal element of the covariance matrix Σ_m. The parameters of the single Gaussian representing the cluster, $g_c^k(X) = \mathrm{N}(X \mid \mu_k, \sigma_k^2)$, is computed as follows:
$$\mu_k(i) = \frac{1}{M_k} \sum_{m=1}^{M_k} E(x_m^{(k)}(i)) = \frac{1}{M_k} \sum_{m=1}^{M_k} \mu_m^{(k)}(i) \quad (10)$$
$$\sigma_k^2(i) = \frac{1}{M_k} \sum_{m=1}^{M_k} E((x_m^{(k)}(i) - \mu_k(i))^2 = \frac{1}{M_k} \sum_{m=1}^{M_k} \sigma_m^{2(k)}(i) + \sum_{m=1}^{M_k} \mu_m^{(k)2}(i) - M_k \mu_k^2(i) \quad (11)$$

Alternatively, we can group the components at the state level using the following distance measure [14]:
$$d(n,m) = -\frac{1}{S} \sum_{s=1}^{S} \frac{1}{P} \sum_{p=1}^{P} \log[b_{ms}(\mu_{nsp})] + \log[b_{ns}(\mu_{msp})] \quad (12)$$

where S is the total number of states in the cluster, P is the number of mixtures per state and $b(.)$ is the observation probability. This method makes it easy to track the state level composition of each cluster. In both cases, the clustering algorithm proceeds as follows:

1) Create one cluster for each mixture up to k clusters.
2) While $k > M_k$, find n and m for which $d(n,m)$ is minimum and merge them.

4. OVERALL SCHEME

Once clustering is complete, it is important to pick the most suitable cluster for feature compensation at each frame. The particle samples are then generated from the representative density of the chosen cluster. Two methods can be explored. The first is to decide the cluster based on the *N*-best transcripts obtained from recognition using multi-condition trained models. Denote the states obtained from the *N*-best transcripts for noisy speech feature vectors at time t as $s_{t1}, s_{t2,\ldots,}s_{tN}$. If state s_{ti} is a member of cluster c_k, we increment $M(c_k)$ by one, where $M(c_k)$ is a count of how many states from the *N*-best list belong to cluster c_k. We choose the cluster based on $\arg\max_k M(c_k)$ and generate samples from it. If more than one cluster satisfies this criterion, we merge their probability density functions. In the second method, we chose the cluster that maximizes the likelihood of the MFCC vector at time t , O_t , belonging to that cluster as follows:

$$C \sim \arg\max_k g_{mc}(O_t \mid C_k) \tag{13}$$

It is important to emphasize here that g_{mc} is derived from multi-condition speech models and has a different distribution from the one used to generate the samples. The relationship between clean clusters and multi-condition clusters is shown in figure 1. Clean clusters are obtained using methods described in section 3. The composition information of these clusters is then used to build a corresponding multi-condition cluster set from multi-condition HMMs. A cluster C_j in clean clusters represents statistical information of a particular section of clean speech. The multi-condition counterpart C_j represents statistics of the noisy version of the same speech section.

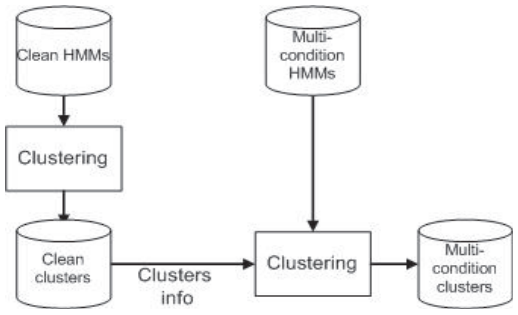

Figure 1. Clustering of multi-condition trained HMMs

Clean clusters are necessary to track clean speech because we need to generate samples from clean speech distributions. However, they are not the best choice for estimating Eq. (13) because the observation is noisy and has a different distribution. The best candidate for computing Eq. (13) is the multi-condition cluster set. It is constructed from multi-condition HMMs that match more closely with noisy speech.

A block diagram of the overall compensation and recognition process is shown in Figure 2. We make inference about the cluster to be used for observation vector $O_t$using both the *N*-best transcripts and Eq. (13) combined together. Samples at frame t are then generated using the pdf of chosen cluster. The weights of the samples are computed using Eq. (3) and compensated features are obtained using Eq. (7). Once the compensated features are available for the whole utterance, recognition is performed again using retrained HMMs with compensated features.

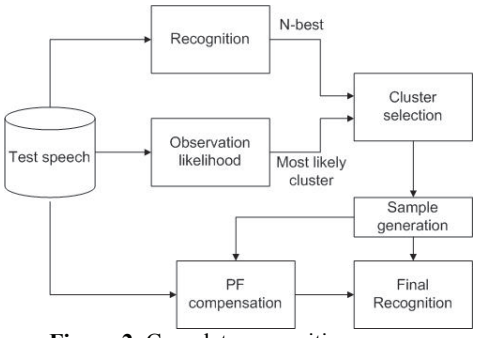

Figure 2. Complete recognition process

5. EXPERIMENTAL RESULTS

To evaluate the proposed framework we experimented on the Aurora 2 connected digit task. We extracted features (39 elements with 13 MFCCs and their first and second time derivatives) from test speech as well as 23 channel filter-bank features thereby forming two streams. One-best transcript was obtained from the MFCC stream using the multi-condition trained HMMs. PFC is then applied to the filter-bank stream (stream two). We chose two clusters, one based on *1*-best and the other selected with Eq. (13).

The multi-condition clusters used in Eq. (13) were from 23 channel fbank features so that the test features from stream two can be directly used to evaluate the likelihood of the observations. For results in these experiments, clusters were formed using method two, i.e., tracking the state-wise composition of each cluster. The number of clusters and particles were varied to evaluate the performance of the algorithm under different settings. From the compensated filter-bank features of stream two, we extracted 39-element MFCC features. Final recognition on these models was done using the retrained HMMs, i.e., multi-condition training data compensated in a similar fashion as described above.

Table 1. Variable number of clusters (100 particles)

Word Accy	20 Clust.	25 Clust.	30 Clust.	MC Trained	Clean Trained
clean	99.11	99.11	99.11	98.50	99.11
20dB	97.76	98.00	97.93	97.66	97.21
15dB	97.00	97.14	96.69	96.80	92.36
10dB	95.21	95.41	93.88	95.32	75.14
5dB	89.48	89.59	87.08	89.14	42.42
0dB	70.16	70.38	68.84	64.75	22.57
-5dB	36.30	36.63	36.94	27.47	NA
0-20dB	89.92	90.10	88.88	88.73	65.94

The results for a fixed number of particles (100) are shown in Table 1. The number of clusters was 20, 25 or 30. To set the specific number of clusters, HMM states were combined and clustering was stopped when the specified number was reached. HMM sets for all purposes were 18 states, with each state represented by 3 Gaussian mixtures. For the 11-digit vocabulary, we have a total of approximately 180 states. In case of, for example, 20 clusters, we have a 9 to 1 reduction of information blocks to choose from for plugging in the PF scheme.

It is interesting to note that best results were obtained for 25 clusters. Increasing the number of clusters beyond 25 did not improve the accuracy. The larger the number of clusters, the more specific speech statistics each cluster contains. If the number of clusters is large, then each cluster encompasses more specific section of the speech statistics. Having more specific information in each cluster is good for better compensation and recognition because the particles can be placed more accurately. However, due to the large number of clusters to choose from, it is difficult to pick the correct cluster for generation of particles. More errors were made in the cluster selection process resulting in degradation in the overall performance.

This is further illustrated in Figure 3. If the correct cluster is known, having large number of clusters and consequently more specific information per cluster will only improve the performance. The results are for 20, 25 and 30 clusters. In the known cluster case, one cluster is obtained using Eq. (13) and the second cluster is the correct one. Correct cluster means the one that contains the state (obtained by doing recognition on the clean version of the noisy utterance using clean HMMs) to which the observation actually belongs to. For the unknown cluster case, the clusters are obtained using Eq. (13) and $1 - best$. It can readily be observed from the known cluster case that if the choice of cluster is always correct, the recognition performance improves drastically. Error rate was reduced by 54%, 59% and 61.4% for 20, 25 and 30 clusters, respectively. Moreover, improvement faithfully follows the number of clusters used. This was also corroborated by the fact that if the cluster is specific down to the HMM state level, i.e., the exact HMM state sequence was assumed known and each state is a separate cluster (total of approximately 180 clusters), the error rate was reduced by as much as 67% [7].

For the results in Table 2, we fixed the number of clusters and varied the number of particles. As we increased the number of particles, the accuracy of the algorithm improves for set A and B combined i.e. for additive noise. The error reduction is 17% over MC trained models. Using a large number of particles implies more samples were utilized to construct the predicted densities of the underlying clean speech features, which is now denser and thus better approximated. Thus, a gradual improvement in the recognition results was observed as the particles increased. In case of Set C, however, the performance was worse when more particles were used. This is so because the underlying distribution is different due to the distortions other than additive noise.

Table 2. Variable number of particles (25 clusters)

	Set A	Set B	Set C	Average
100 particles	90.02	91.03	89.26	90.1
500 particles	90.03	91.10	89.07	90.07
1000 particles	90.02	91.13	89.07	90.07
MC Trained	88.41	88.82	88.97	88.73
Clean Trained	64.00	67.46	65.39	65.73

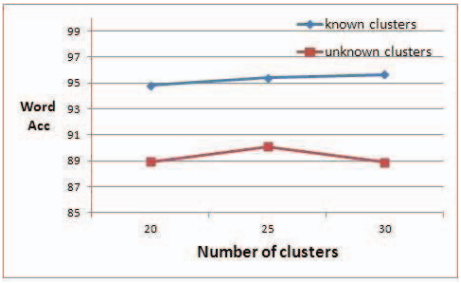

Figure 3. Accuracy when correct cluster known vs. unknown

6. SUMMARY

We have proposed an Integrated PF-HMM approach where we incorporate statistical information available from the HMMs, to make up for the lack of suitable state transition model for speech signals required for particle filters. This enables us to use the PF framework to compensate noisy speech signals. We further developed a scheme to merge statistically similar information in HMM states to enable us to find the right section of HMMs to dynamically plug in the particle filter algorithm. Results show that if we use information from HMMs that match specifically well with section of speech being compensated, significant error reduction is possible compared to multi-condition HMMs.

11. REFERENCES

[1] J.-L. Gauvain and C.-H. Lee, "Maximum a posteriori estimation for multivariate observations of Markov chains," *IEEE Trans. Speech Audio Processing*, vol. 2, no. 2, pp.291-299, Apr.1994.

[2] C. Leggetter and P. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models," *Computer Speech and Language*, vol. 9, pp.171-185, 1995.

[3] P. J. Moreno, "Speech recognition in noisy environments," *PhD Thesis*, Department of ECE, Carnegie Mellon Univ., 1996.

[4] 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.

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

[6] 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.

[7] A. Mushtaq, Y. Tsao and C.-H. Lee, "A Particle Filter Compensation Approach to Robust Speech Recognition." *Proc. Interspeech*, 2009.

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

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

[10] R. G. Brown and P. Y.-C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering, 3rd edition*, Prentice Hall, 1996.

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

[12] A. Doucet and A. M. Johansen, "A tutorial on particle filtering and smoothing: Fifteen years later," *Technical Report*, 2008. [Online]. http://www.cs.ubc.ca/~arnaud/doucet_johansen_tutorialPF.pdf

[13] T. Watanbe, K. Shinoda, K. Takagi, and E. Yamada, "Speech recognition using tree-structured probability sensity function," in *Proc. Int. Conf. Speech Language Processing '94*, 1994, pp. 223-226.

[14] S. J. Young, J. J. Odell, and P. C. Woodland, "Tree-based state tying for high accuracy acoustic modeling, " *Proc. ARPA Human Language Technology Workshop*, pp. 307–312, 1994.