A Particle Filter Compensation Approach to Robust LVCSR

Duc Hoang Ha Nguyen*, Aleem Mushtaq§, Xiong Xiao†, Eng Siong Chng*†, Haizhou Li*†‡ and Chin-Hui Lee§
*School of Computer Engineering, Nanyang Technological University, Singapore
†Temasek Lab@NTU, Nanyang Technological University, Singapore
‡Institute for Infocomm Research, A*STAR, Singapore
§School of Electrical and Computer Engineering, Georgia Institute of technology, Atlanta, USA
ng0008ha@e.ntu.edu.sg, aleem@gatech.edu, xiaoxiong@ntu.edu.sg,
aseschng@ntu.edu.sg, hli@i2r.a-star.edu.sg, chl@ece.gatech.edu

***Abstract*—We extend our previous work on particle filter compensation (PFC) to large vocabulary continuous speech recognition (LVCSR) and conduct the experiments on Aurora-4 database. Obtaining an accurately aligned state and mixture sequence of hidden Markov models (HMMs) that describe the underlying clean speech features being estimated in noise is a challenging task for sub-word based LVCSR because the total number of triphone models involved can be very large. In this paper, we show that by using separate sets of HMMs for recognition and compensation, we can simplify the models used for PFC to a great extent and thus facilitate the estimation of the side information offered in the state and mixture sequences. When the missing side information for PFC is available, a large word error reduction of 28.46% from multi-condition training is observed. In the actual scenarios, an error reduction of only 5.3% is obtained. We are anticipating improved results that will narrow the gap between the system today and what's achievable if the side information could be exactly specified.**

Index Terms: speech feature compensation, particle filter, robustness, clustering

I. Introduction

State-of-the-art automatic speech recognition (ASR) systems perform well when there is a good match between training and testing conditions. The accuracy degrades in adverse conditions as the acoustic mismatch increases. One way to alleviate this problem is to adapt the models according to the specific environment of interest. Maximum a posteriori (MAP) [1], maximum linear likelihood regression (MLLR) [2] and parallel model combination (PMC) [3] are instances of this approach. Another way is to compensate the features by removing the distortion effects corrupting clean speech. The aim, in this case, is to map them to the feature space used in the training phase. Vector Taylor series (VTS) is an instance of this approach [4], and has also been adopted for model adaptation [5]. Cepstral mean subtraction (CMS) [6], cepstral mean variance normalization (CVN) [7] and ETSI advanced front-end (AFE) [8] are other notable examples of feature compensation.

Particle filters were initially used to track noise information in noisy signals to subsequently obtain compensated clean features [9][10][11]. Here, the noise was treated as a state variable while speech was considered as the signal corrupting the observation noise, and a Taylor Series approximation was used to approximate the clean speech signal by applying a minimum mean square error (MMSE). Being a Monte Carlo method, particle filters are versatile and can handle a broad category of dynamical systems not constrained by linearity and Gaussianity requirements that inhibit Kalman Filter [12] and extended Kalman Filter [13]. Particle filter compensation (PFC) [14][15] algorithms compensate noisy speech features by directly tracking the clean speech features in the spectral domain. The recognition is performed on mel-frequency cepstral coefficient (MFCC) features extracted from the newly estimated filter bank features.

Despite the versatility of particle filters, a state transition model that adequately captures the dynamic properties of the speech signal is still required. Due to the nature of speech, it is extremely difficult to find such a model. PFC alleviates the problem by introducing information from HMMs trained with clean speech to propagate the particles. Typical HMMs have many states that hold the statistical information of all variations in the speech corpus of interest. It is a challenging task to select the proper state from the complete HMM set to plug in the PFC algorithm. The difficulty is increased for the large vocabulary systems because the number of triphone HMMs used to model these systems can be very large and exceed $10,000$.

To overcome this problem, we exploit the feature of PFC where the HMM set used at the front end for compensation and the HMM set used for recognition at the back-end can be separate. Consequently, the HMM set that is integrated within the particle filter framework can be much simpler and consist of a small number of states. Starting from approximately 1600 tied-states (or physical states), the number of statistical information units is reduced to less than 10 by first stripping the triphone models to mono-phone models and then clustering them to shrink the number of states to the desired level.

The PFC algorithm is tested on the Aurora-4 large vocabulary continuous speech recognition task. It is shown that a large error reduction of 28.46% is achieved with 120 clusters if the side information is accurately known. Similarly good performance is maintained (error reduction of 20.66% and 19.97% respectively) even when fewer number of states such as 10 clusters and 5 clusters are used. However, in actual

scenarios, the best error reduction achieved is only 5.3% and that is with 3 clusters. As the number of clusters is increased, errors made in cluster selection increase and the performance degrades. We are exploring algorithms that will improve the cluster selection process and bring the real environment performance closer to the known cluster case.

II. Background

HMMs are widely used for estimating the likelihood of an observed feature vector. In this study, we will now look at an HMM as a generator of some feature vectors. In other words, is it possible to generate feature vector samples from an HMM? Fig. 1 shows the example of the digit two, sampled using an HMM that was trained with 45 handwritten $twos$ [16]. The top row shows some of the actual handwritten digits used to train the HMM, while the bottom row shows the digits generated using the HMM. It can be noted that all the curves traced by the human hand to write a two have been captured by the HMM and subsequently reproduced in the artificial two. The models potential for sample generation is apparent. HMM as a sample generator can also be justified based on the fact that, when modeling a signal with an HMM, it is an underlying assumption that the signal is generated from such a model.

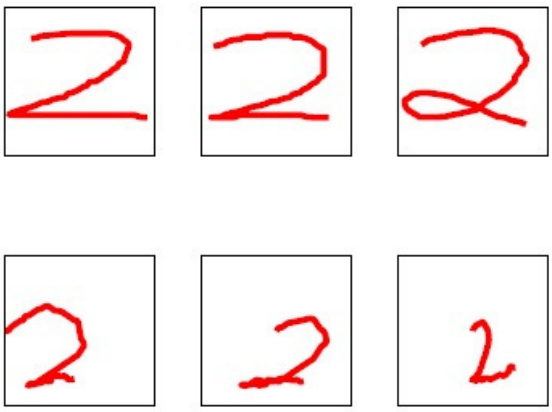

Fig. 1. An example of on-line handwritten digit

The sampling process will comprise of two steps. First, a state will be picked based on the state transition probabilities. Once an appropriate state has been selected, samples will be generated from its observation probability. If an HMM is intended to be used for sample generation, its observation distributions should preferably be easy to sample from. Any appropriate sampling scheme can be used to sample from the HMMs observation distribution.

HMMs differ in nature from the standard tracking algorithms and by themselves, have limited capability for tracking a continuously varying signal. Both HMMs and PF have states, but these states different in nature. The state of a PF is a real quantity. On the contrary, the states of an HMM may be used only as a modeling strategy. The observation distribution of an HMM, however, is not only a real quantity, but also a valid source for sample generation. Consequently, there is a possibility of utilizing the observation distribution to generate the samples in the PF algorithm. In such a setup, the observation distribution of the HMM will correspond to the state of the PF. The structure can be viewed as a three layer scheme as shown in Fig. 2.

The red line is the observed signal, the blue line is the state of the signal being estimated and S_1, S_2 and S_3 in the circle are the HMM states, whose observation distribution is used to generate the samples representing the state. Instead of obtaining the samples from the state space model as is done in a conventional PF algorithm, the samples are generated from the observation model of a particular state of an HMM. The weights of the samples can then be computed using the observed signal. The diameter of the sample in the figure indicates its weight. The idea will be actualized for tracking of speech signals contaminated by noise in the next chapter.

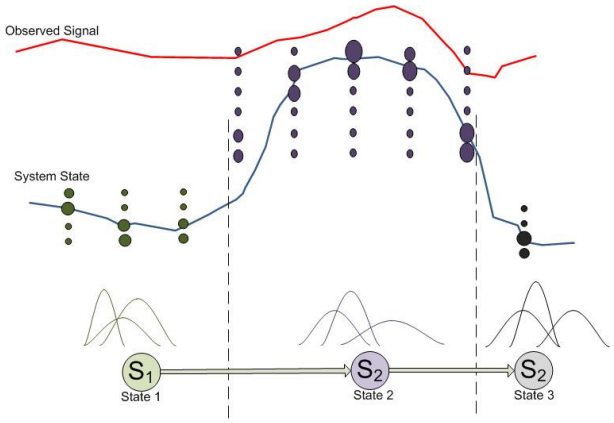

Fig. 2. HMM for sample generation

III. Overview of PFC

A. Distortion Model

As in [5], if the clean speech, x, is corrupted by an additive noise, n, and a distortion channel, h, then we can represent the noise corrupted speech, y, as in Fig. 3. Assuming known statistics of the noise parameters,

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

where $y = log(S_y(m_p))$, $x = log(S_x(m_p))$ and $h = log(|H(m_p)|^2)$ and $S(m_p)$ denotes the p^{th} mel spectrum.

$$S_y(m_p) = S_x(m_p)|H(m_p)|^2 + S_N(m_p). \tag{2}$$

We utilize the distortion model to evaluate weights of clean feature samples in PFC algorithm and will be presented later.

B. The Compensation Scheme

The compensation scheme is illustrated in Fig. 4. The compensation process requires background information together with additional side information which can be provided by a

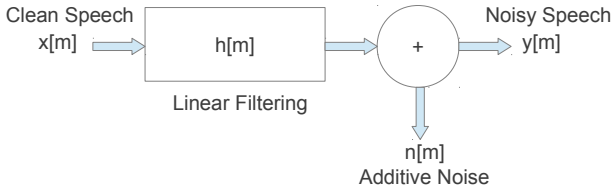

Fig. 3. Distortion model

decoder. The background information include clean acoustic model (or clean HMMs) and noise model. The side information is a set of nuisance parameters, Φ. Similar to *stochastic matching* [17], we can iteratively find Φ followed by decoding as

$$\Phi' = \arg\max_{\Phi} p(Y'|\Phi, \Lambda), \tag{3}$$

where Y' is the noisy or compensated utterance.

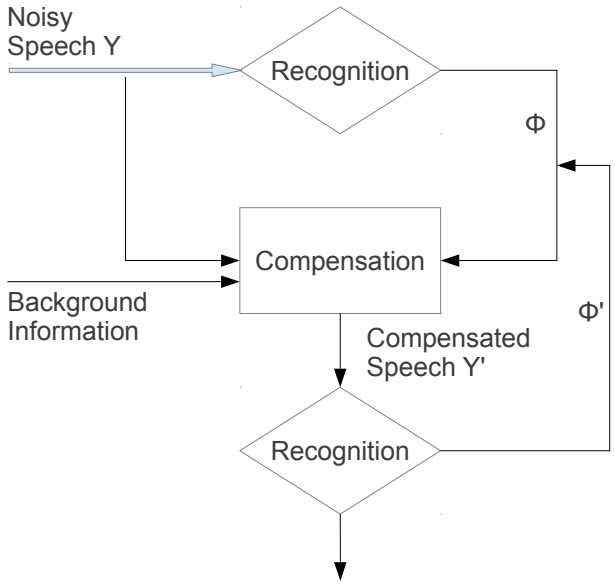

Fig. 4. Compensation scheme based on Stochastic Matching

The parameters Φ in Equation (3) in our particle filter compensation (PFC) scheme, correspond to the corresponding correct HMM state sequence and mixture component sequence. These sequences provide critical information for density approximation of clean features in PFC. The PFC is briefly summarized in next subsection.

C. A Brief Summary of PFC

Speech tracking using PFC is summarized as follows [14]:

1) Posterior density of speech, based on the current observation, is represented by a finite number, N_s, of support points,

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

where x_t^i for $i = 1, ..., N_s$ are the support points of PF and $\delta()$ denotes the Dirac delta function.

2) The weight vector, w_t^i, associated with the support points, approximates the posterior density and are determined based on the concept of *importance sampling* [18] with:

$$w_i^t = w_{t-1}^i \frac{p(y_t|x_t^i)p(x_t^i|x_{t-1}^i)}{q(x_t^i|x_{t-1}^i, y_t)} \tag{5}$$

3) PFC is done in the spectral domain. Given additive noise statistics with no channel effects [5], we can obtain $p(y|x)$ using the distortion model (1) as

$$\begin{aligned} p(y|x) &= F'(u) \\ &= p(u)\frac{e^{y-x}}{e^{y-x}-1} \end{aligned} \tag{6}$$

where $F(u)$ and $p(u)$ are the Gaussian cumulative function and Gaussian function with noise mean, μ_n, and noise variance, σ_n^2, and

$$u = \log(e^{y-x} - 1) + x \tag{7}$$

4) The density $q(x_t^i|x_{t-1}^i, y_t)$ plays a crucial role in particle filtering. Known as the *importance sampling density*, it is used to generate the particle samples. The distribution is obtained by clustering HMMs as described in next section[15]. The sampling density then becomes

$$q(x_t^i|x_{t-1}^i, y_t) = \sum_{k=1}^{K} m_{k,C_t} \mathcal{N}(x_t^i; \mu_{k,C_t}, \Sigma_{k,C_t}) \tag{8}$$

where m_{k,C_t}, μ_{k,C_t} and Σ_{k,C_t} are the weight, mean and variance of the mixture k in cluster C_t.

5) Finally, the compensated features are estimated as [14]:

$$x_t = \sum_{i=1}^{N_s} w_t^i x_t^i \tag{9}$$

IV. A Clustering Approach to Obtaining Correct HMM Information

Placing the samples at the right locations plays a critical role in the performance of the particle filter. In PFC, these locations are derived from the statistical information contained in the HMM states. If the HMM state chosen for this placement is the correct one, the subsequent estimation of the underlying clean speech density will be accurate. Otherwise, the density estimate will be erroneous. The selection of the correct state is difficult when there is a large number of states to chose from. To overcome this problem, we merge the states into clusters. The total number of clusters can be much less than the number of states, therefore, 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 [19]:

$$d(m,l) = \int g_m(x) \log \frac{g_m(x)}{g_l(x)} dx + \int g_l(x) \log \frac{g_l(x)}{g_m(x)} dx$$
$$= \sum_i [\frac{\sigma_m^2(i) - \sigma_l^2(i) + (\mu_l(i) - \mu_m(i))}{\sigma_l^2(i)}$$
$$+ \frac{\sigma_l^2(i) - \sigma_m^2(i) + (\mu_l(i) - \mu_m(i))}{\sigma_m^2(i)}], \tag{10}$$

where $\mu_m(i)$ is the $i-th$ element of the mean vector μ_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, $g_k(X) = \mathcal{N}(X; \mu_k, \sigma_k^2)$, is computed as follows:

$$\mu_k(i) = \frac{1}{M_k} \sum_{m=1}^{M_k} E(x_m^{(k)}(i)) = \frac{1}{M_k} \sum_{m=1}^{M_k} \mu_m^{(k)}(i)$$
$$\sigma_k^2(i) = \frac{1}{M_k} \sum_{m=1}^{M_k} E(x_m^{(k)}(i) - \mu_k(i))^2$$
$$= \frac{1}{M_k} \left(\sum_{m=1}^{M_k} \sigma_m^{2(k)}(i) + \sum_{m=1}^{M_k} \mu_m^{2(k)}(i) - M_k \mu_k^2(i) \right) \tag{11}$$

Alternatively, we can group the components at the state level using the following distance measure [20]:

$$d(m,l) = -\frac{1}{P} \sum_{p=1}^{P} (\log[b_m(\mu_{lp})] + \log[b_l(\mu_{mp})]) \tag{12}$$

where P is the number of mixtures per state and $b(.)$ is the observation probability. The clustering algorithm proceeds as follows:

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

Once clustering is complete, it is important to pick the most suitable cluster for feature compensation at each frame. Samples can be generated from the Gaussian mixture density, representing the selected cluster, using conventional Monte Carlo methods. Selecting the best possible cluster is the single most important factor effecting the performance of the PFC algorithm. One approach for this selection is to derive the cluster information from the $N - best$ transcripts obtained from recognition done using multi-condition trained models. Alternatively, we can also chose the cluster that maximizes the likelihood of the MFCC vector at time t, O_t, belonging to that cluster as follows:

$$C = \arg\max_k g_{mc}(O_t|C_k), \tag{13}$$

where $g_{mc}(.)$ represents the probability that O_t corresponds to the cluster C_k.

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. Clean clusters are obtained using methods described above. 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 the clean speech. The multi-condition counterpart C_j represents statistics of the noisy version of the same speech section.

V. Simple vs Complex Models

In the PFC algorithm, the compensation is done at the front end of the ASR system. Consequently, the HMM set used for compensation (Box 1 of Fig. 5) and the one used for recognition (Box 2 of Fig. 5) can be different and independent of one another. This relaxation can be exploited in the overall compensation and recognition processes. For the compensation phase, simpler models are better since the states are ultimately merged into clusters that represent diverse statistics.

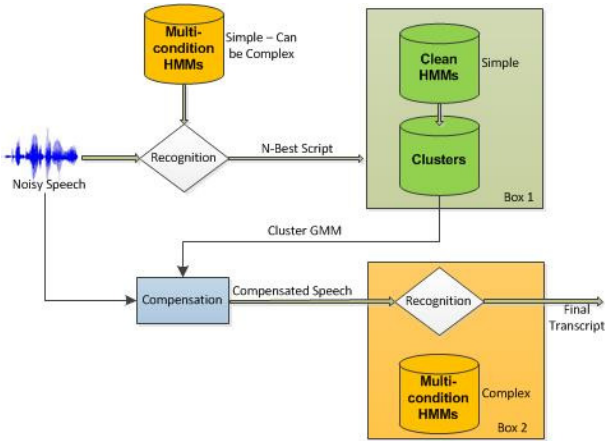

Fig. 5. Simple vs complex models

Starting from complex HMMs does not give a significant advantage in the clustering phase and thus the compensation phase because the statistical information related to a specific speech segment will be lost at some stage. On the contrary, complex models are much more useful for the recognition phase. Here, the objective is to obtain precise information about the speech segment being evaluated. Complex models capture specific speech segments statistical information better.

It must be noted that if precise information about the speech segmented being compensated is available, the compensation will improve. However, there will be a greater risk of selecting wrong statistics, i.e., the state might not represent the speech being compensated. It has been observed that the penalty incurred by wrong choice of cluster/state overwhelms the advantage gained from using complex and specific models and therefore, simpler models work better in the compensation phase. Next section presents empirical analysis of this issue.

VI. PFC for LVCSR

In the PFC algorithm, four HMM sets are used in various roles. The roles of these models are explained next. The most

important aspect of PFC, aside from the observation model, is the placement of the samples. Clean FBANK HMM set (hereafter known as Set 1) is used to generate the samples because clean speech is being estimated from these samples, and clean HMMs provide the distributions that best represent the clean speech statistics. These models are derived from FBANK features because PFC is done in the FBANK domain.

It is critical that the correct model from the HMM set is chosen for the treatment of a particular frame so that the samples can be generated from a distribution that precisely represents the underlying speech for that frame. The structure of the Set 1 HMMs should therefore be such that it is easy to pick the most suitable model at each frame. As is described in section V, a large number of models makes this selection harder. For LVCSR systems, subword acoustic models are a popular choice and triphone representation achieves the best recognition performance. However, in the case of PFC, the large number of models required in the triphone representation make the model selection problem even harder.

Monophone models provide a convenient solution to the problem. Although, accuracy of the statistical representation is compromised for the case of monophone models compared to the triphone model, but the number of statistical units is drastically reduced by a ratio of approximately $1:20$. By further clustering the monophone models into 10 or less statistical units, the composition of the set is reduced to about $1:250$ when compared to the triphone models. This procedure simplifies the cluster selection process to a great extent, but the task of estimating the appropriate cluster from noisy speech is another complication. Set 1 is unsuitable for the task because:

1) It is built from FBANK features, which have inferior discrimination capability compared to MFCC features.
2) Clean models perform poorly in the recognition task when applied to noisy speech.
3) Monophone models can not compete with the triphone models in the recognition task.

To overcome this complication, a second set of HMMs (Set 2) is deployed to obtain speech information from the noisy signal. This set is derived with the aim of getting optimum recognition performance. Hence, the HMMs in set 2 are triphone models built using multi-condition MFCC features.

A. Alignment of set 1 and set 2

As the HMMs in Set 2 are used to select the appropriate cluster from HMMs in Set 1, a good alignment between the two sets is essential to obtain good performance with PFC algorithm. The two sets, however, use different features, structures (one is made up of monophone while the other of triphone models) and data (one uses clean and the other uses noisy speech). Consequently, the two sets can be severely misaligned. To overcome this problem, the clean MFCC HMMs (Set 3) are used as the source and both Set 1 and Set 2 are derived from it. The technique for this alignment procedure is explained in Fig. 6.

We train Set 1 HMMs in 2 steps. Step 1 computes forward and backward probabilities using clean MFCC monophone

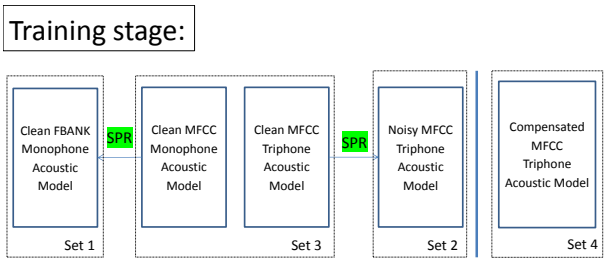

Fig. 6. *A block diagram illustrates training process using the single-pass retraining (SPR).*

HMMs on clean MFCC features. Step 2 estimates parameters of FBANK monophone HMMs using the statistics information from Step 1, together with clean FBANK features. This is known as single-pass retraining [21]

In this way, the state/phone alignment (i.e., the posterior component probabilities) used to estimate parameters of monophone FBANK HMMs is the same as one generated by using the monophone MFCC HMMs. Therefore, same component label of two states in two different feature domain will model the same sound but in two different feature domain.

Training HMMs in Set 2 is similar. Step 1 compute forward and backward probabilities using triphone HMMs on clean MFCC features. Step 2 estimates HMM parameters using the statistics from Step 1 along with noisy MFCC features.

Since all HMM parameters in Sets 1 and 2 are estimated based on state alignment computed from clean MFCC HMMs, a state mapping between the two sets can be obtained by just using the same component labels.

B. Models for Compensated Features

As described in Section V, the HMMs (Set 4) used in the final recognition of the compensated data is isolated from the compensation process. Therefore, Set 4 is independent of sets 1,2 and 3. Set 4 is trained using multi-condition training data that has been compensated like we would process the test data in actual scenario. Since there are no constrains on these models, their complexity can be increased to the optimum level needed to obtain the best possible recognition performance.

VII. Aurora-4 Experiments

In the following we present PFC experiments on the Aurora-4 task. We focus on training simple and complex models used in PFC, an oracle experiment to estimate the upper bound of the method and an actual experiment to evaluate the performance of the system.

A. General Configurations

The hidden Markov model toolkit (HTK) [20] was used to extract speech features and train acoustic models. Log mel filter bank (FBANK) coefficients (23 coefficients) were extracted from 16KHz sampled speech signals and enhanced by PFC method. Mel-frequency cepstral coefficients (13 coefficients) and their first and second differential features are then extracted from compensated FBANK and used as speech

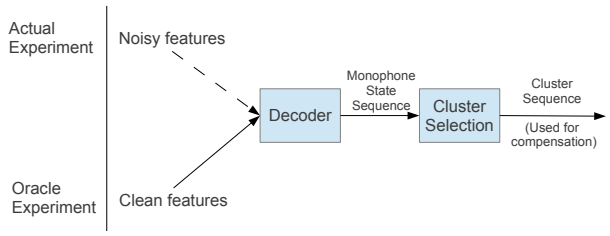

Fig. 7. *A block diagram illustrates oracle experiment vs. actual experiment.*

features for speech recognizer. Ceptral mean normalization was also applied to reduce the channel mismatch. A bigram language model was used with language model scale factor set to 15.

The four acoustic models were trained as described earlier. In this study, HMMs in Set 1 have 120 states with 3 Gaussian mixtures per state. The complexity of HMMs in Set 2, 3 and 4 were the same and have 1594 tied-states with 16 Gaussian mixtures per state.

In the testing phase, we are interested in additive background noises. Six noisy test sets (car, babble, restaurant, street, airport and train noises) without channel mismatch were used to evaluate the PFC performance. The noise statistics are estimated from silence frames of each utterance.

As PFC works in the FBANK domain, the compensated FBANK features are then transformed to MFCC domain by DCT transformation. For dynamic features (delta and delta-delta features), we have two options: re-compute the dynamic features from the compensated MFCC features or just use the original noisy dynamic features. We will discuss about the two options in more details in the next sections.

B. Experiments with Oracle Cluster ID

To estimate the potential of PFC, we first build an oracle experiment with high accuracy of cluster selection. In this experiment, we utilize the stereo data in Aurora-4 to generate oracle state sequence which is clean state sequence and used as noisy state sequence and thus the cluster selection is exact (see Fig. 7). In this way, we can focus on optimizing particle sampling and evaluate the upper bound of the PFC method.

Oracle experiments of clustering PFC is then investigated. Un-clustered FBANK monophone HMMs has 120 states and denoted by "set 1-120". We group 120 states into 10 (or 5, 3, 2, 1) clusters as discussed in the previous sections and denote as "Set 1-10" (or 1-5, 1-3, 1-2, 1-1 respectively).

The word accuracies of these versions of Set 1 are shown in Table I. In the study, 120 is the largest cluster count used. Although, the count can be increased to 1594, which is the starting number of states if clustering directly from triphone acoustic model, and it will most likely improve the performance beyond the best figure of 85.6% because the statistical information is more precise. However, it hasn't been explored due to the fact that obtaining good side information in case of such a large number of clusters will be nearly impossible in real scenarios.

TABLE I

Word accuracy (%) obtained by PFC using oracle cluster ID information. Dynamic features are recomputed from PFC compensated features. 6 types of noisy environments are shown (2-car, 3-babble, 4-restaurant, 5-street, 6-airport, 7-train).

No. of clusters	Noisy Test Cases						
	2	3	4	5	6	7	Avg.
-	87.4	81.5	75.6	78.4	80.9	75.4	79.9
1	86.6	82.6	76.2	79.3	80.7	76.2	80.3
2	87.2	83.9	78.2	80.4	82.1	77.1	81.5
3	87.3	84.5	78.9	81.3	82.3	79.0	82.2
5	88.1	84.9	81.2	83.0	84.0	82.1	83.9
10	88.2	85.8	81.3	83.5	83.7	81.7	84.0
120	88.8	86.3	83.4	84.4	87.1	83.8	85.6

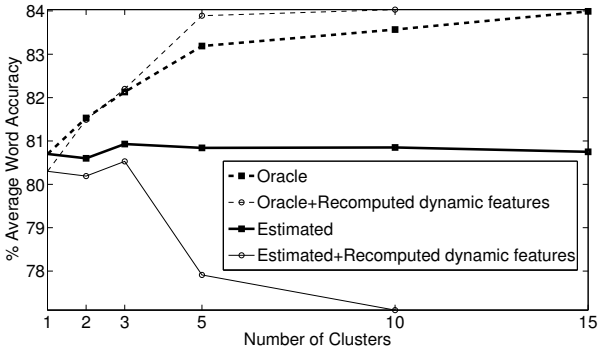

Fig. 8. Performance of PFC with different numbers of clusters. Both PFC with oracle cluster ID and PFC with estimated cluster ID are shown.

On the other side, 1-cluster is the smallest cluster count that can be used. Apart from the fact that the performance for this case improves over the baseline multi-condition training, the setup has it's own advantages. First, the estimation of side information is not required, making the compensation process very efficient. Secondly, with 1-cluster, no errors can be made in the estimation of side information and therefore, the actual performance and the oracle performances are the same.

C. Experiments with Estimated Cluster ID

Now we investigate PFC using estimated side information, i.e. the cluster IDs. A cluster ID sequence is generated by using 1-best cluster selection method presented in Section IV. The overall performance is shown in Fig. 8. From the figure, we have two major observations. First, when oracle cluster IDs are used, the performance of PFC improves monotonically with the number of clusters. However, when estimated cluster IDs are used, the performance of PFC peaks at 3 clusters, and then degrades when more and more clusters are used. This observation shows that only when accurate cluster information are available (e.g. in the case of oracle cluster ID), PFC will benefit from the more detailed side information provided by more clusters. In practice, the gain of more detailed side information is offset by the wrong estimated cluster ID and hence the performance of PFC will decrease.

The second observation from Fig. 8 is that whether to re-compute the dynamic features from PFC compensated static features plays an important role in the overall performance

TABLE II
Word accuracy (%) obtained by PFC using estimated cluster IDs and WITH re-computed the dynamic features.

No. of clusters	Noisy Test Cases						
	2	3	4	5	6	7	Avg.
2	87.2	82.3	76.3	79.4	79.5	76.4	80.2
3	87.3	82.8	76.6	79.3	79.5	77.7	80.5
5	86.1	78.5	72.6	78.0	75.5	76.8	77.9
10	86.0	76.9	71.5	78.0	74.6	75.6	77.1

TABLE III
Word accuracy (%) obtained by PFC using estimated cluster IDs and WITHOUT re-computing the dynamic features.

No. of clusters	Noisy Test Cases						
	2	3	4	5	6	7	Avg.
2	88.1	82.9	76.0	79.1	81.1	76.6	80.6
3	88.4	82.4	76.7	79.5	81.4	77.2	80.9
5	88.7	81.8	76.2	79.5	81.8	76.9	80.8
10	88.5	82.1	76.7	79.4	82.0	76.6	80.8
15	88.8	81.9	76.5	79.0	81.9	76.4	80.8

of the PFC framework, especially when estimated cluster IDs are used. If dynamic features are not re-computed and when estimated cluster IDs are used, the performance of PFC is quite stable when more than 3 clusters are used. However, if dynamic features are re-computed, the PFC performance degrades quickly as more than 3 clusters are used. The observation is different when oracle cluster IDs are used. This suggests that the dynamic features is very sensitive to the errors in cluster ID estimation. A possible explanation is that when wrong cluster is used, the temporal structure of the PFC compensated static features are seriously distorted, hence the re-computed dynamic features will be also wrong. This suggests a possible way to improve the PFC framework is to enforce the correlation between adjacent frames in a more explicit way.

The detailed recognition word accuracies of PFC with estimated cluster ID are shown in Table II and Table III. The best result of 80.9% is obtained with 3 clusters and do not re-compute dynamic features. This represents a 5.3% relative error rate reduction over the multi-condition baseline system (79.9%).

VIII. Summary and Future Work

We have extended the PFC framework to LVCSR and tested it on the Aurora-4 task. An incorrect state selection issue caused by a big triphone set in LVCSR can be lessened with a clustering approach. However, there is a trade-off in choosing the number of clusters. With less clusters, there is a less risk of incorrect selection; but with more clusters, the more precisely side information will be provided to the PFC if it can be correctly estimated. The performance gap between the oracle and actual experiments is still rather large. Hence, more studies are required in the future to narrow the gap. Note that the strategy of cluster selection is important, we should continue pursuing for a better strategy in cluster selection.

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] C.J. Leggetter and P.C. 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] M.J.F. Gales and S.J. Young, "Robust continuous speech recognition using parallel model combination," *Speech and Audio Processing, IEEE Transactions on*, vol. 4, no. 5, pp. 352–359, 1996.

[4] 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.

[5] 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.

[6] 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.

[7] Olli Viikki and Kari Laurila, "Cepstral domain segmental feature vector normalization for noise robust speech recognition," *Speech Communication*, vol. 25, no. 1, pp. 133–147, 1998.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

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

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

[14] 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.

[15] 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.

[16] Christopher M Bishop, "Pattern recognition and machine learning (information science and statistics)," 2007.

[17] A. Sankar and C.-H. Lee, "A maximum-likelihood approach to stochastic matching for robust speech recognition," *Speech and Audio Processing, IEEE Transactions on*, vol. 4, no. 3, pp. 190–202, 1996.

[18] 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.

[19] K. Shinoda and C.-H. Lee, "A structural bayes approach to speaker adaptation," *Speech and Audio Processing, IEEE Transactions on*, vol. 9, no. 3, pp. 276–287, 2001.

[20] S. Young, G. Evermann, M.J.F. Gales, T. Hain, D. Kershaw, G. Moore, J.J. Odell, D. Ollason, D. Povey, V. Valtchev, and P.C. Woodland, "The HTK Book," Tech. Rep., University of Cambridge, 2004.

[21] M.J.F. Gales, *Model-Based Techniques for Noise Robust Speech Recognition*, Ph.D. thesis, Cambridge University, 1995.