Microphone Distance Adaptation Using Cluster Adaptive Training for Robust Far Field Speech Recognition

Animesh Prasad and Khe Chai Sim

National University of Singapore, Singapore

{animesh,simkc}@comp.nus.edu.sg

1. Introduction

- **Motivation**
  - DNNs have become the state-of-the-art for acoustic modeling.
  - However, there exists a big performance degradation if the acoustic conditions of the testing data are very different from that of the training data.
  - We investigate the case of adapting DNN to accommodate varying speaker distances to the microphone.
  - Most existing techniques require additional setup, assume the position of speaker to be invariant during an utterance and have little scope in terms of frame level adaptation.
- **Main Goal**
  - This scenario tries to imitate meeting transcription task using a single microphone, where speakers can be at variable distances from the microphone.
- **Gain**
  - 0.9% absolute improvement over multi-style trained model.

2. Cluster Adaptive Training

- **Formulation**
  - In CAT each or some of the hidden layers of different bases are combined as:

$$z^i(\mathbf{x}) = \sum_{k=1}^{K} \lambda_k \sigma\left(W_k^i z^{i-1}(\mathbf{x}) + b_k^i\right) \quad (1)$$

$$\hat{z}^i(\mathbf{x}) = \sigma\left\{\sum_{k=1}^{K} \lambda_k \left(W_k^i z^{i-1}(\mathbf{x}) + b_k^i\right)\right\} \quad (2a)$$

$$= \sigma\left(\hat{W}^i z^{i-1}(\mathbf{x}) + \hat{b}^i\right) \quad (2b)$$

where,

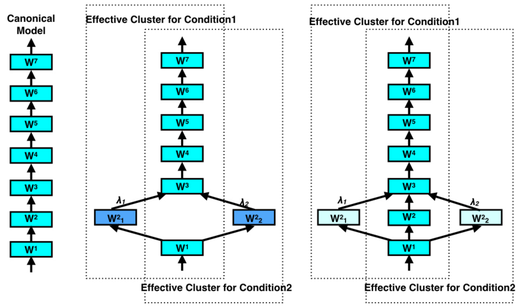
$$\hat{W}^i = \sum_{k=1}^{K} \lambda_k W_k^i$$

$$\hat{b}^i = \sum_{k=1}^{K} \lambda_k b_k^i$$

  - CAT uses DNNs as multiple bases of a canonical parametric space.
  - In our case different distances from the microphone form natural clusters for the basis.

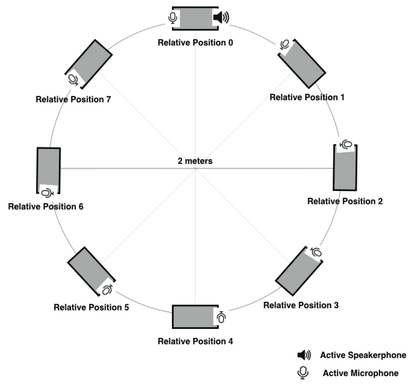
3. Approach

- **CAT for Distance Adaptation**
  - DI-DNN module trained on pooled multiple condition data acts as the initialization for the CAT-DNN system
  - The weights for DD modules (W_c^i) are learned only by the data from distance condition c. This is done by switching off all other DD modules when a frame from distance condition c is feed forward
  - After the DD modules converge the λ_c and DD modules are updated in alternate epochs.
  - Finally, the whole network is fine-tuned with small learning rate. The final set of λ_c is discarded.
  - The interpolation weights (λ_c) of these DD modules can be estimated per utterance or per condition in supervised manner or unsupervised manner (by using the canonical model to generate pseudo transcripts)



4. Experiments

- **Varying Distance WSJ0**
  - Data duplication of WSJ0 at various distance in the practical limit of meeting scenario.
  - Each of the iPads played $\frac{1}{8}$th of the training data and $\frac{1}{8}$th of the testing data while all other iPads recorded the audio, in the setting as shown.
  - Each relative position has 7138 utterances by 83 speakers to be used as training and development set and 330 separate utterances by 12 speakers in testing set.



- **Experimental Setup**
  - Canonical DNN configuration: 7 layers, 1024 nodes per layer
  - Features: 39 dimensional MFCC, splicing context window of 5, global CMVN transform on train set and per utterance CMVN transform on test set
  - Training: CE criteria, with per layer discriminative pre-training initialization with dropouts
  - Tools: Kaldi and CNTK
- **CAT Recipe**
  - While fine-tuning the DD module the canonical module is frozen; and while re-estimating the interpolation parameters whole network is frozen.
  - The interleaving update for DD modules and interpolation parameters is done for 2 epochs each with learning rate of 0.05, halved after each epoch.
  - Whole network is fine tuned with small learning rate for single iteration in the end.
  - During testing unsupervised adaptation is performed by taking the hypothesis and alignment from the decoding of the canonical model. The parameters are initialized to sum to 1 and learned per utterances with learning rate of 0.01.

5. Results

- **Baselines**
  - WER from varying distance from speaker to microphone shows the performance degradation as the distance increases.

Model \ Test	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	Avg
M_0	19.1	20.5	25.9	30.0	30.5	31.0	29.2	29.0	**26.9**
M_4	24.1	23.3	25.4	26.0	26.4	25.9	26.1	25.7	**25.4**

  - WER on multi-style trained models.

Model \ Test	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	Avg
M_{04}	20.7	21.0	25.2	27.4	27.9	27.2	27.1	26.7	**25.4**
M_{0246}	22.8	22.1	25.4	25.9	25.7	25.2	25.9	24.8	**24.7**

- **CAT Results**
  - WER for CAT on single layer without the canonical module. Applying CAT on the layers close to input show higher gain.

Model \ Test	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	Avg
L_1	19.9	20.7	25.2	26.5	27.2	26.9	27.0	26.5	**25.0**
L_2	20.3	20.9	25.5	26.4	27.5	26.8	27.2	26.4	**25.1**
L_3	20.5	21.2	25.6	26.8	27.4	26.8	27.4	26.6	**25.3**
L_4	20.5	21.3	25.7	27.2	27.3	26.9	27.5	26.7	**25.4**

  - WER for CAT on single layer with the canonical module. Applying CAT on the layers close to input show higher gain. Gain is higher as compared to models without canonical module.

Model \ Test	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	Avg
L_1	19.5	20.3	24.9	26.3	26.8	26.7	26.9	26.3	**24.7**
L_2	19.9	20.5	25.0	26.5	27.0	26.8	27.1	26.5	**24.9**
L_3	20.2	21.1	25.3	26.7	27.2	26.7	27.3	26.5	**25.1**
L_4	20.3	21.2	25.5	27.0	27.1	26.9	27.4	26.6	**25.2**

  - WER for CAT on multiple layers with the canonical module. The gain by applying CAT on multiple layers stack diminishingly.

Model \ Test	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	Avg
L_{1-2}	19.4	20.2	24.8	26.2	26.6	26.5	26.7	26.3	**24.6**
L_{1-3}	19.3	20.1	24.7	26.2	26.6	26.4	26.7	26.3	**24.5**
L_{1-4}	19.4	20.2	24.7	26.3	26.6	26.5	26.7	26.3	**24.6**

6. Conclusions

- VD-WSJ0 can be used to investigate distance adaptation in interesting meeting scenarios.
- CAT presents a valid framework for ad hoc varying distance adaptation.
- Absolute WER gain of 1.3% in seen and 0.7% in unseen conditions resulting in average gain of 0.9%