

Feature compensation in the cepstral domain employing model combination



Source : Speech communication

Author : Wooil Kim and John H.L. Hansen

Reporter:邱聖權

Professor:陳嘉平



Introduction

- This paper introduces a cepstral feature compensation by model combination.
- GMMs of clean speech and noise are used for model combination to generate the noisy speech model.



Feature compensation via parallel combined Gaussian mixture model

- The relationship between the cepstral feature vectors of clean speech \mathbf{x} , additive noise \mathbf{n} and noise-corrupted speech \mathbf{y} , is presented as follows:

$$\mathbf{y} = \mathbf{x} + \mathbf{C} \log(\mathbf{I} + \exp(\mathbf{C}^{-1}(\mathbf{n} - \mathbf{x}))) = \mathbf{x} + g(\mathbf{x}, \mathbf{n})$$

where \mathbf{C} and \mathbf{C}^{-1} denotes the DCT and its inverse transform.

GMMs of clean speech and noise



- The clean speech model in the cepstral domain is estimated as a GMM through training on the clean speech database .
- The noise model is estimated as a single Gaussian model using the silence duration of the incoming speech or noise samples off-line.



Feature compensation via parallel combined Gaussian mixture model

- The mean of noisy speech model can be expressed as:

$$\boldsymbol{\mu}_y = E\{\boldsymbol{x}\} + E\{g(\boldsymbol{x}, \boldsymbol{n})\} = \boldsymbol{\mu}_x + \boldsymbol{r}$$

\boldsymbol{r} is the bias.

- Applying to each Gaussian component:

$$\boldsymbol{\mu}_{y,k} = \boldsymbol{\mu}_{x,k} + \boldsymbol{r}_k$$

- The mean and covariance of clean speech and noise model in cepstral domain are transformed to the log-spectral domain using an inverse DCT:

$$\boldsymbol{\mu}^{\log} = \boldsymbol{C}^{-1} \boldsymbol{\mu}$$

$$\boldsymbol{\Sigma}^{\log} = \boldsymbol{C}^{-1} \boldsymbol{\Sigma} (\boldsymbol{C}^{-1})^T$$



Parallel model combination

$$Y_i^{\log}(\tau) = F(X_i^{\log}(\tau), N_i^{\log}(\tau)) = \log(\exp(X_i^{\log}(\tau)) + g \cdot \exp(N_i^{\log}(\tau)))$$

$Y_i^{\log}(\tau)$, $X_i^{\log}(\tau)$, $N_i^{\log}(\tau)$, g denote the i th element of the clean speech, noise, noise - corrupted speech, and the gain respectively in the log - spectral domain.



Computing noisy speech model by PMC

- Transforming from log spectral domain to linear spectral domain:

$$\mu_i^{\text{lin}} = \exp(\mu_i^{\text{log}} + \Sigma_{ii}^{\text{log}} / 2)$$

$$\Sigma_{ij}^{\text{lin}} = \mu_i^{\text{lin}} \mu_j^{\text{lin}} (\exp(\Sigma_{ij}^{\text{log}}) - 1)$$

- Computing noisy speech model parameters:

$$\mu_y^{\text{lin}} = \mu_x^{\text{lin}} + g \cdot \mu_n^{\text{lin}}$$

$$\Sigma_y^{\text{lin}} = \Sigma_x^{\text{lin}} + g^2 \cdot \Sigma_n^{\text{lin}}$$

- Transforming from linear spectral domain to log spectral domain:

$$\mu_i^{\text{log}} \approx \log(\mu_i^{\text{lin}}) - \frac{1}{2} \log\left(\frac{\Sigma_{ii}^{\text{lin}}}{(\mu_i^{\text{lin}})^2} + 1\right)$$

$$\Sigma_i^{\text{log}} \approx \log\left(\frac{\Sigma_{ii}^{\text{lin}}}{\mu_i^{\text{lin}} \cdot \mu_j^{\text{lin}}} + 1\right)$$



Feature compensation

- Reconstruction of clean speech by MMSE

$$\hat{\mathbf{x}}_{\text{MMSE}} = \int \mathbf{x} p(\mathbf{x} | \mathbf{y}) d\mathbf{x} = \int (\mathbf{y} - g(\mathbf{x}, \mathbf{n})) p(\mathbf{x} | \mathbf{y}) d\mathbf{x}$$
$$\cong \mathbf{y} - \sum_{k=1}^K \mathbf{r}_k p(k | \mathbf{y})$$

Block diagram of the PCGMM-based feature compensation method

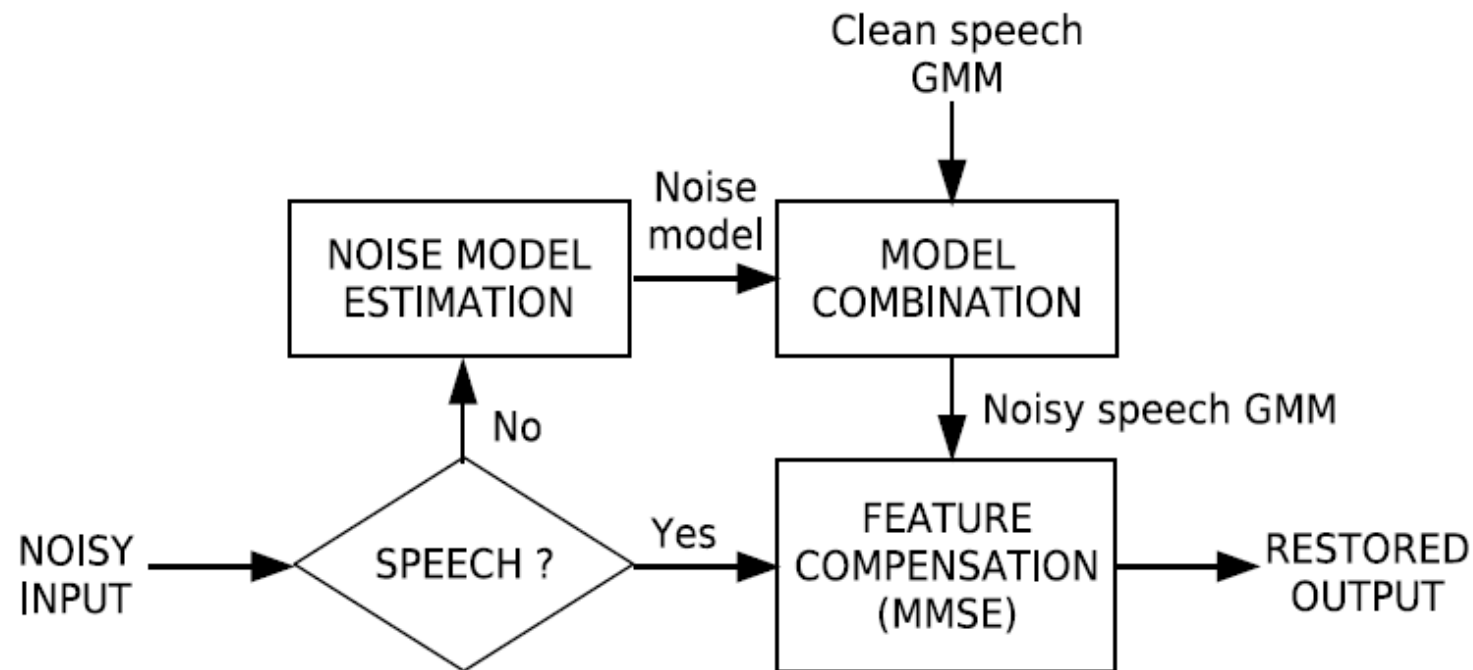
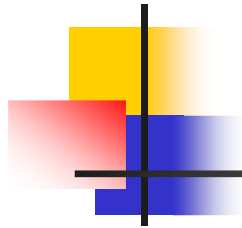


Fig. 1. Block diagram of the PCGMM-based feature compensation method.



Properties of this method

- This method does not require an additional training procedure using a noise-corrupted speech database.
- The noise model is estimated as a single Gaussian model.
- The GMMs are accomplished in the cepstral domain.



PCGMM-based feature compensation employing multiple environmental models

- Model adaptation can be applied in order to address the time-varying background noise.
- In order to reduce the computation complexity, utilizing multiple models estimated off-line can be effective for compensating input features adaptively under time-varying noisy conditions and eliminating the need for online model combination.

Interpolation of multiple models

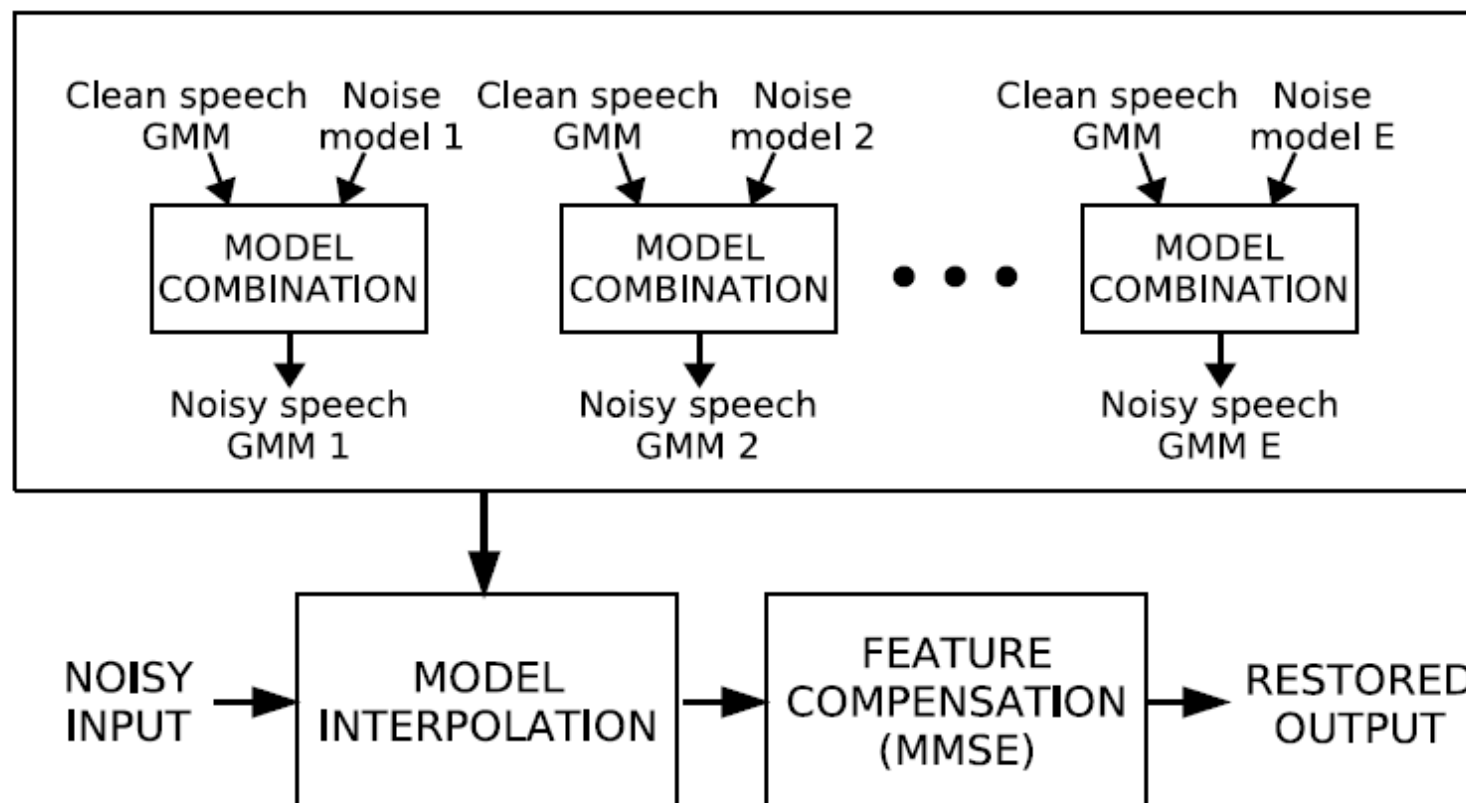


Fig. 2. PCGMM-based method employing the interpolation of multiple models.

Interpolation of multiple models

- The a posteriori probability of each possible environment is estimated over the incoming noisy speech.

Given the incoming noisy speech feature vectors $\mathbf{Y}_t = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_t]$ the sequential posterior probability of a specific environment GMM G_i among E models over the input speech feature \mathbf{Y}_t is :

$$p(G_i | \mathbf{Y}_t) = \frac{p(G_i)P(\mathbf{Y}_{t-1} | G_i)p(\mathbf{y}_t | G_i)}{\sum_{e=1}^E p(G_e)P(\mathbf{Y}_{t-1} | G_e)p(\mathbf{y}_t | G_e)}$$



The MMSE estimated feature

- The clean feature at frame t is reconstructed using the interpolated compensating terms as follows

$$\hat{\mathbf{x}}_{t,\text{MMSE}} \cong \mathbf{y}_t - \sum_{e=1}^E p(G_e | \mathbf{Y}_t) \sum_{k=1}^K \mathbf{r}_{e,k} p(k | G_e, \mathbf{y}_t)$$



Computational reduction via sharing components

- Reducing the computational complexity by sharing the statistically similar components among the multiple environment models.
- The Kullback–Leibler distance is used to represent the separation between multi-component GMMs.

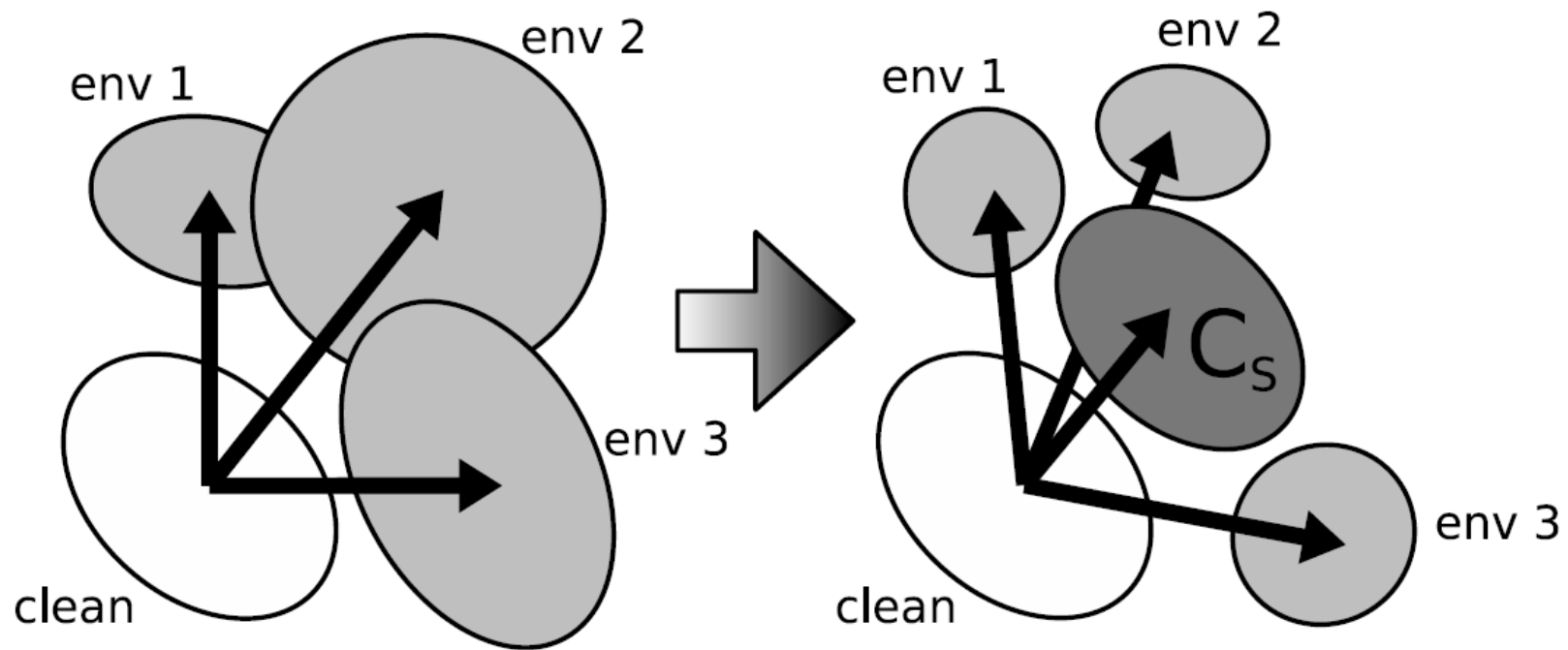
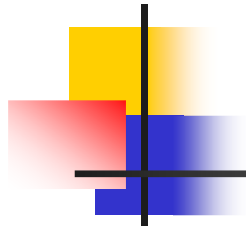


Fig. 3. Illustration of PDF mixture sharing.



Algorithm of sharing GMMs

- **D** is the set of distances between Gaussian components, and **C_S** is the set of shared Gaussian components:

- **Step 0:** $\mathbf{D} = \{d_1, d_2, \dots, d_K\}$, $\mathbf{C}_S = \emptyset$

$$d_k = \sum_{e=2}^E kl_dist(g_{1,k}, g_{e,k}), \quad 1 \leq k \leq K. \quad (21)$$

- **Step 1:** $\hat{k} = \arg \min_k d_k \in \mathbf{D}$.
- **Step 2:** $\mathbf{C}_S = \mathbf{C}_S \cup \{\hat{k}\}$, $\mathbf{D} = \mathbf{D} - \{d_{\hat{k}}\}$.
- **Step 3:** if $N(\mathbf{C}_S) = K_S$, then stop, else go back to **Step 1**.

Mean and variance of shared components

- The parameters of the merged Gaussian components which are shared are computed as follows:

$$\boldsymbol{\mu}_{y,k}^S = \frac{1}{E} \sum_{e=1}^E \boldsymbol{\mu}_{y,e,k}, k \in C_s$$

$$\boldsymbol{\Sigma}_{y.k}^S = \frac{1}{E} \sum_{e=1}^E \left(\boldsymbol{\Sigma}_{y,e,k} + (\boldsymbol{\mu}_{y,e,k} - \boldsymbol{\mu}_{y,k}^S)(\boldsymbol{\mu}_{y,e,k} - \boldsymbol{\mu}_{y,k}^S)^T \right)$$



Experimental results of Aurora2(clean training)


	SetA	SetB	SetC	Average
Baseline	58.56	56.67	66.16	59.32
SS	66.08	62.07	75.91	66.44
CMN	61.65	66.76	62.30	63.82
SS + CMN	73.65	77.00	74.84	75.23
PMC	81.04	81.45	76.86	80.37
AFE	85.77	84.40	84.60	84.99



Experimental results of Aurora2(clean training)

	SetA	SetB	SetC	Average
PCGMM	84.29	82.34	72.18	81.09
PCGMMm	85.48	84.51	81.20	84.24
PCGMMmv	79.44	78.91	82.30	79.80
FCLS1	78.90	78.64	75.64	78.14
FCLS2	83.52	84.01	76.52	82.32
VTs	75.80	77.53	76.95	76.72

	SetA	SetB	SetC	Average
PCGMMm + SS	85.70	84.28	84.61	84.91
PCGMMm + SS + CMN	87.21	86.03	87.18	86.73
FCLS2 + SS + CMN	85.71	86.29	80.47	84.89
VTs + SS + CMN	81.06	83.75	83.48	82.62



Experimental results of Aurora2(clean training)

	SetA	SetB	SetC	Average
IM-PCGMM	85.13	83.49	70.97	81.64
IM-PCGMM + SS	85.76	83.55	80.84	83.89
IM-PCGMM + SS + CMN	87.17	85.49	85.14	86.09

	SetA	SetB	SetC	Average
IM-PCGMM + SS + CMN	87.17	85.49	85.14	86.09
IM-PCGMM32 + SS + CMN	86.46	85.11	84.44	85.52
IM-PCGMM64 + SS + CMN	85.57	84.41	83.45	84.68