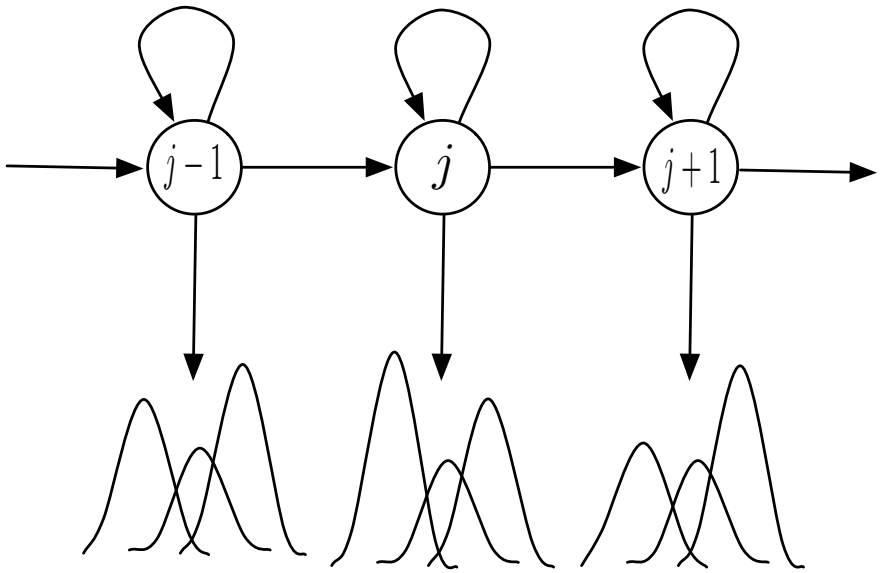
Joint Uncertainty Decoding with Unscented Transform for Noise-robust Subspace Gaussian Mixture Models

Liang Lu, Arnab Ghoshal, Steve Renals
University of Edinburgh

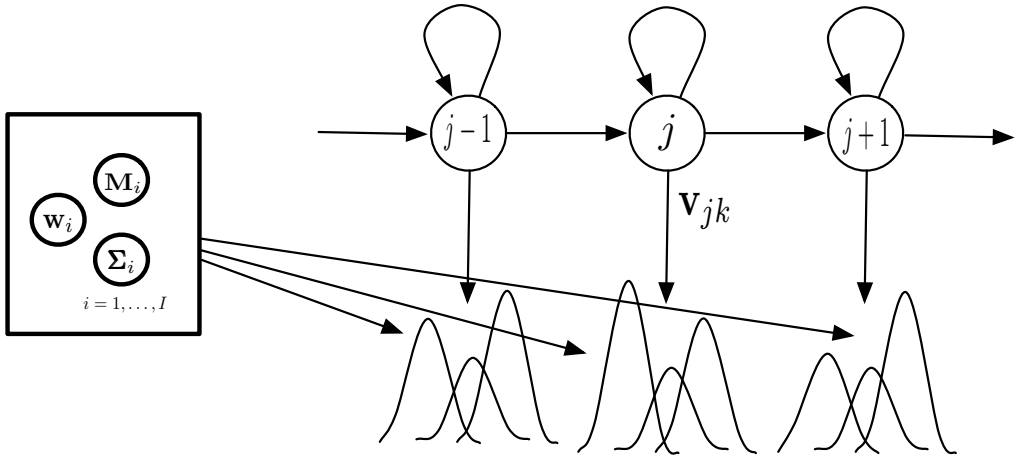
Outline

- Motivation
  - Subspace GMM (SGMM) works well in matched speech condition [Povey et al., 2011]
  - In mismatched condition (i.e. noise), the gain disappears
- Goal
  - Noise compensation for SGMM
- Method
  - Model space compensation
  - Joint uncertainty decoding (JUD)
  - Vector Taylor series
  - Unscented transform

HMM-GMM acoustic model



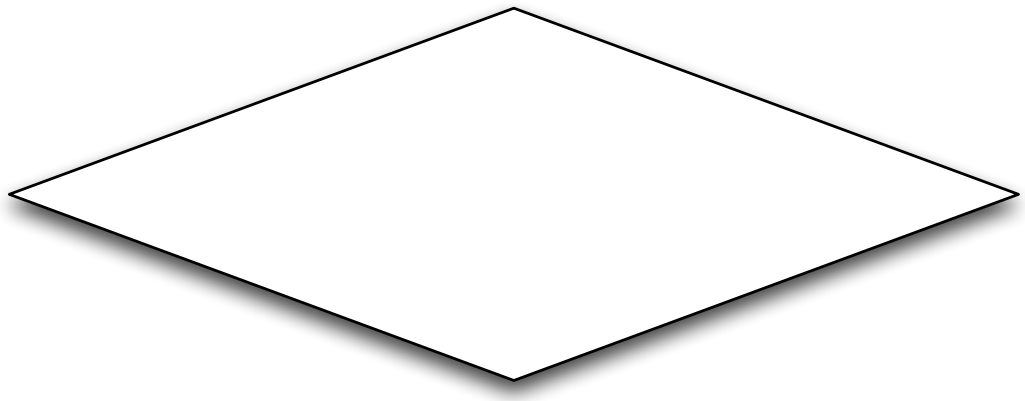
Subspace Gaussian Mixture Models [Povey et al., 2011]



- Global
  - $\mathbf{M}_i$ is the basis for means
  - $\mathbf{w}_i$ is the basis for weights
  - $\mathbf{\Sigma}_i$ is the covariance matrix
- State-dependent
  - $\mathbf{v}_{jk}$ is low dimensional vectors (e.g. 40dim)

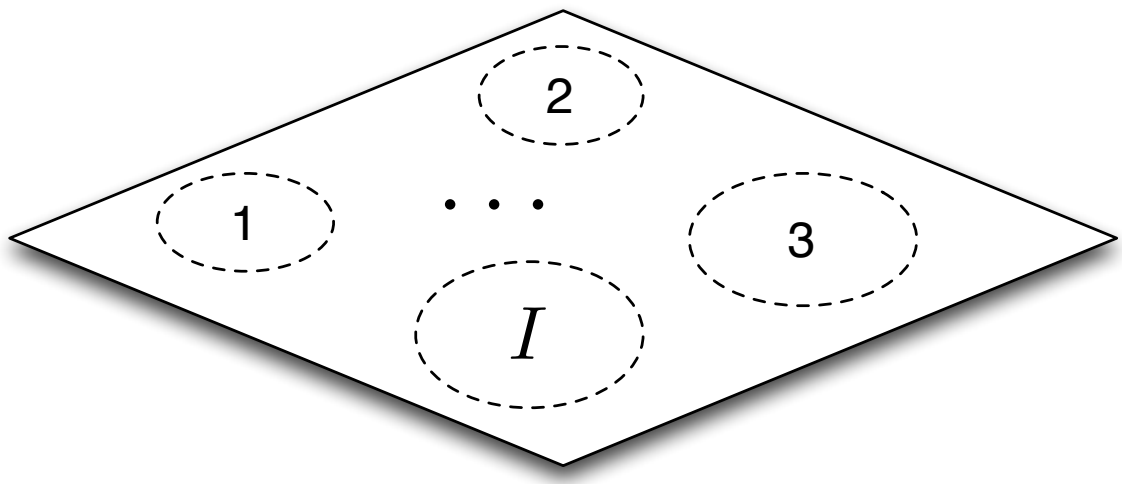
Subspace Gaussian Mixture Models

- More intuitively, suppose we have an acoustic space like this



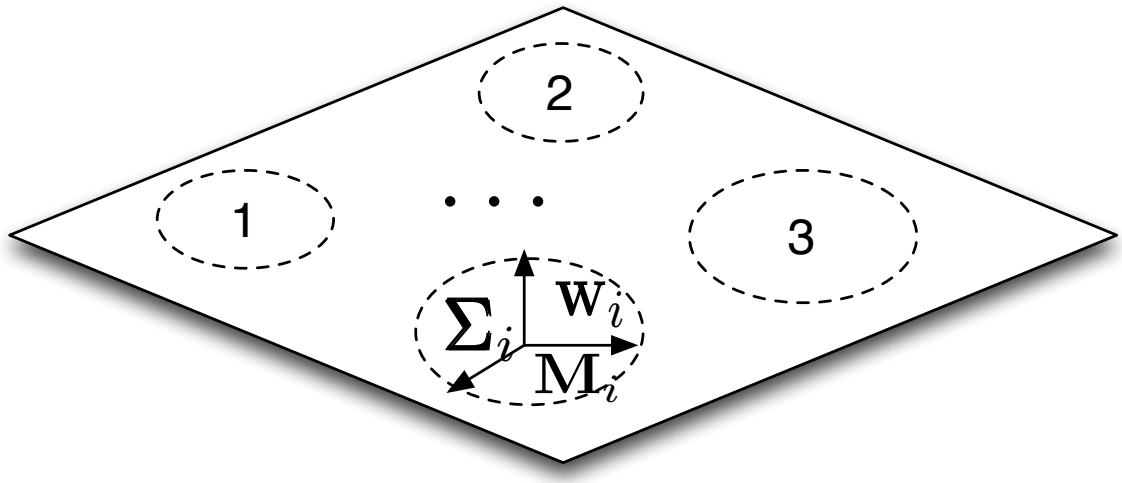
Subspace Gaussian Mixture Models

- We then partition the whole acoustic space into I regions e.g. $I = 400$
- This can be done by learning a GMM using the training data



Subspace Gaussian Mixture Models

- We then introduce some parameters to structure each region



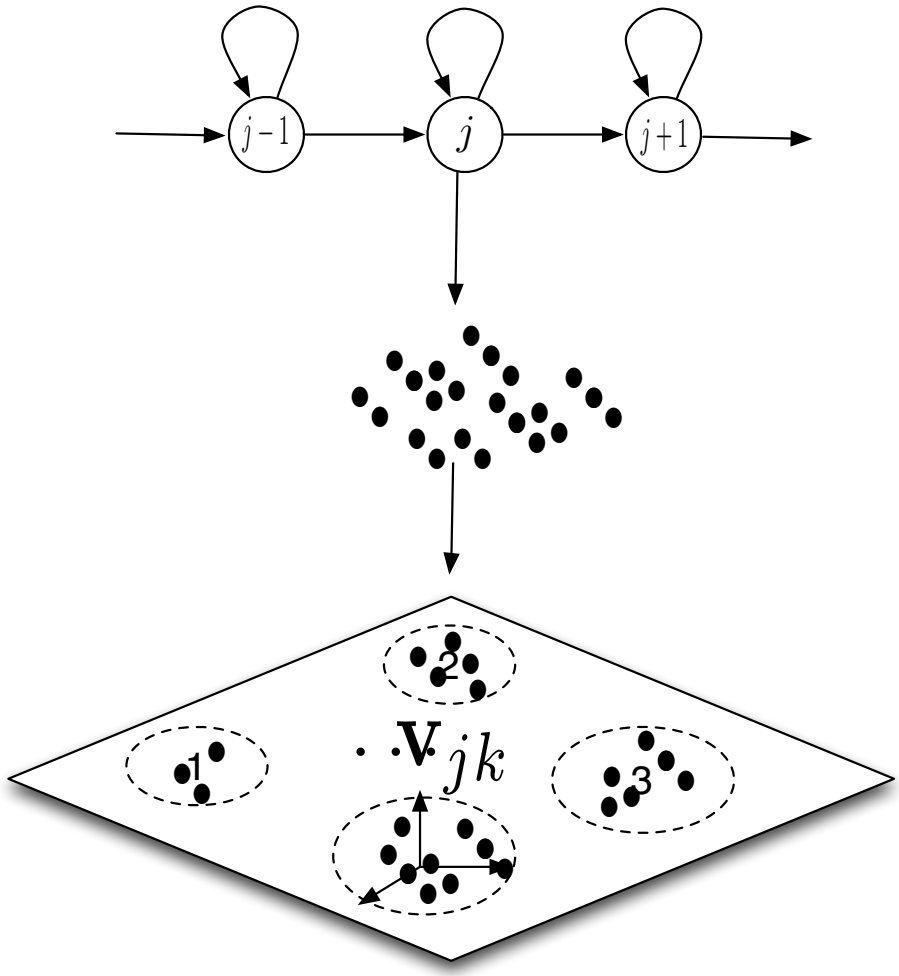
Σ_i - model the covariance of this region

$\mathbf{M}_i$ - span the basis for Gaussian mean

$\mathbf{w}_i$ - span the basis for Gaussian weight

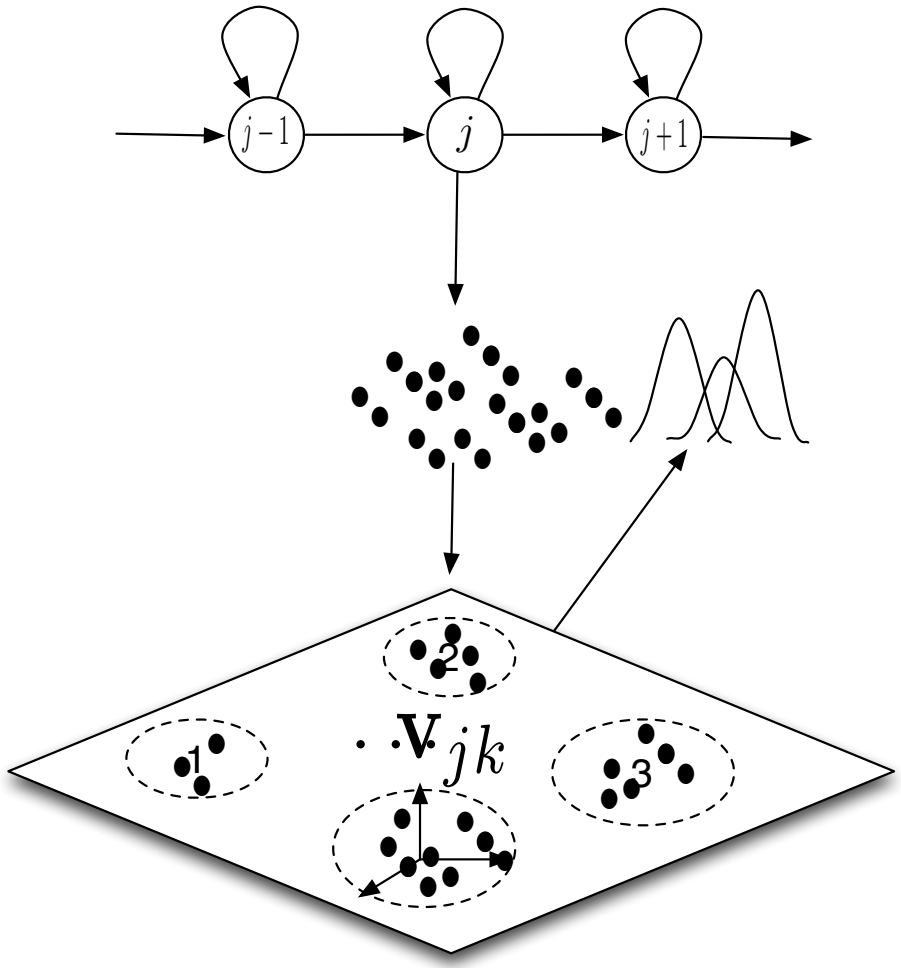
Subspace Gaussian Mixture Models

Given a class with some data, such as an HMM state



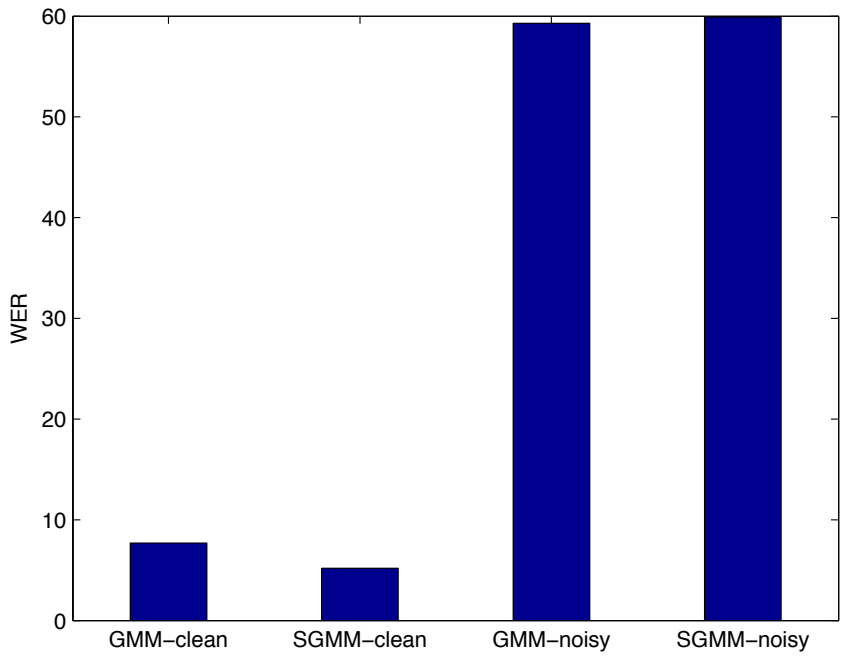
Subspace Gaussian Mixture Models

Then we learn a GMM for this class



Noise compensation

- Larger modelling power $\rightarrow$ higher recognition accuracy.
  - Our systems on Aurora 4, the #Gaussians is 6.4M (SGMM), vs. 50k (GMM).
  - SGMM vs. GMM $\rightarrow$ 5.2% vs. 7.7% on clean condition
  - SGMM vs. GMM $\rightarrow$ 59.9% vs. 59.3% on noisy condition
- Can we do noise compensation for SGMMs ?



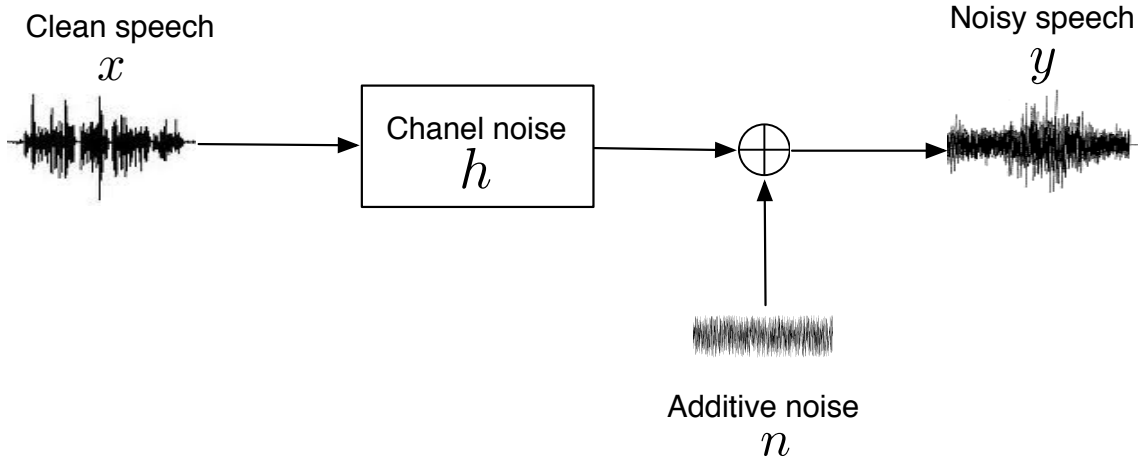
Noise compensation

There are numerous work on noise compensation for robust ASR

- Feature domain
  - Spectral subtraction, cmn/cvn
  - Cepstral mean square error estimation
  - Algonquin
  - Splice
  - Feature space vector Taylor series (VTS)
- Model domain
  - MLLR, noise constraint MLLR
  - PMC, Data-driven PMC (DPMC), iterative DPMC
  - VTS, joint uncertainty decoding (JUD)
  - Linear spline interpolation (LSI)
  - Unscented transform (UT)
- Hybrid
  - Noise adaptive training

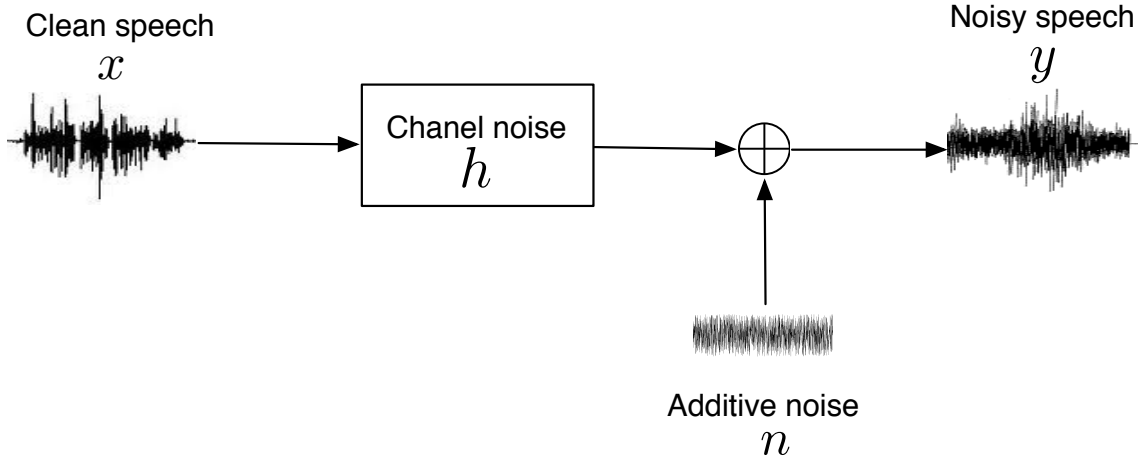
Noise compensation for SGMM

- Model space compensation for SGMM
- Not data-driven but using heuristic knowledge
- Mismatch function $\mathbf{y} = f(\mathbf{x}, \mathbf{h}, \mathbf{n}, \boldsymbol{\alpha})$ [Acero, 1990]
- $\boldsymbol{\alpha}$ denotes the phase term between noise and speech [Deng et al., 2004].



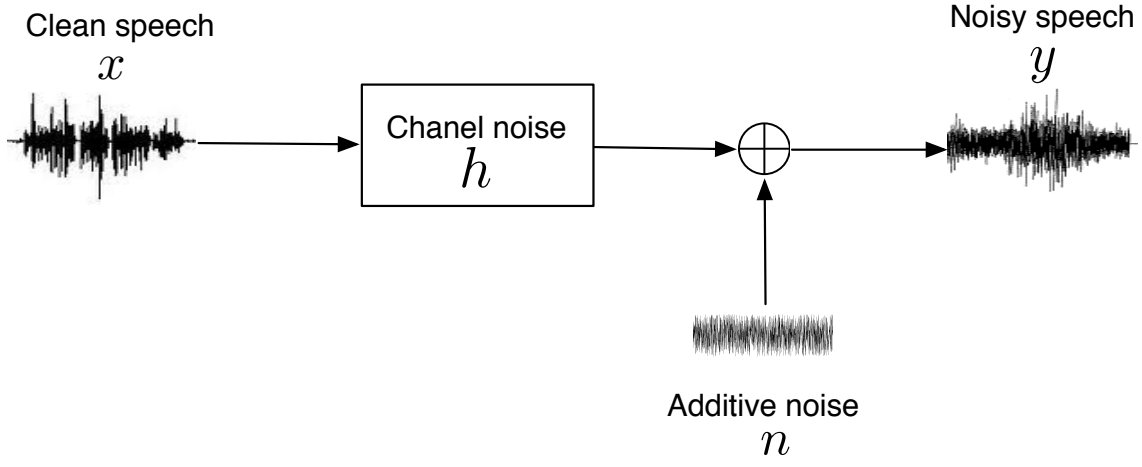
Noise compensation for SGMM

- Model space compensation for SGMM
- Not data-driven but using heuristic knowledge
- Mismatch function $\mathbf{y} = f(\mathbf{x}, \mathbf{h}, \mathbf{n}, \boldsymbol{\alpha})$ [Acero, 1990]
- $\boldsymbol{\alpha}$ denotes the phase term between noise and speech [Deng et al., 2004].



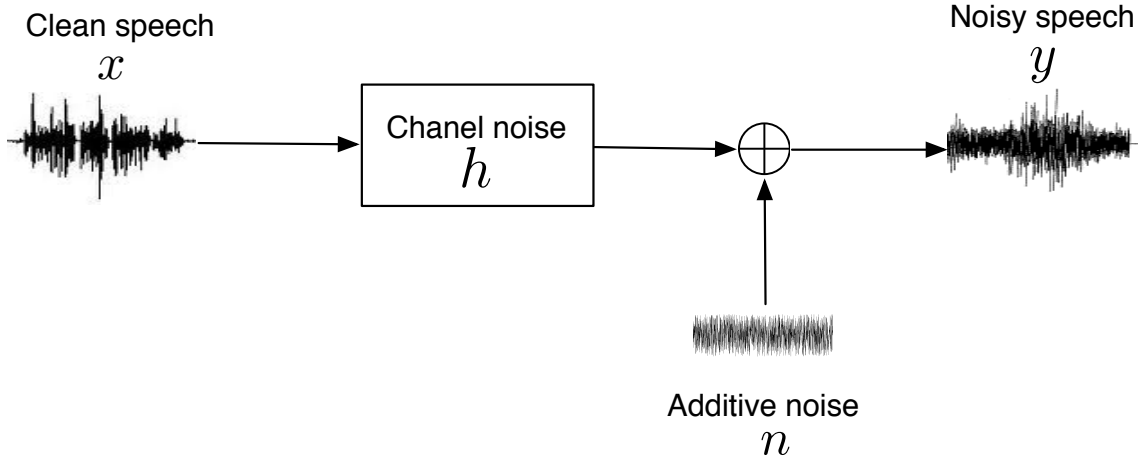
Noise compensation for SGMM

- Model space compensation for SGMM
- Not data-driven but using heuristic knowledge
- Mismatch function $\mathbf{y} = f(\mathbf{x}, \mathbf{h}, \mathbf{n}, \boldsymbol{\alpha})$ [Acero, 1990]
- $\boldsymbol{\alpha}$ denotes the phase term between noise and speech [Deng et al., 2004].



Noise compensation for SGMM

- Model space compensation for SGMM
- Not data-driven but using heuristic knowledge
- Mismatch function $\mathbf{y} = f(\mathbf{x}, \mathbf{h}, \mathbf{n}, \boldsymbol{\alpha})$ [Acero, 1990]
- $\boldsymbol{\alpha}$ denotes the phase term between noise and speech [Deng et al., 2004].



Noise compensation for SGMM

The mismatch function is

$$\begin{aligned}\mathbf{y} &= f(\mathbf{x}, \mathbf{h}, \mathbf{n}, \boldsymbol{\alpha}) \\ &= \mathbf{x} + \mathbf{h} + \mathbf{C}\log\Big[\mathbf{1} + \exp\left(\mathbf{C}^{-1}\left(\mathbf{n} - \mathbf{x} - \mathbf{h}\right)\right) \\ &\quad + \underbrace{2\boldsymbol{\alpha} \bullet \exp\left(\mathbf{C}^{-1}(\mathbf{n} - \mathbf{x} - \mathbf{h})/2\right)}_{\textit{phase term}}\Big]. \end{aligned} \tag{1}$$

where $\mathbf{C}$ be the DCT matrix.

Noise compensation

- Aim: estimate $\boldsymbol{\mu}_y$ and $\boldsymbol{\Sigma}_y$ for each Gaussian component.
- Difficulty: $\mathbf{y} = f(\mathbf{x}, \mathbf{h}, \mathbf{n}, \boldsymbol{\alpha})$ is highly nonlinear, no analytic solution!
- Solution:
  - Vector Taylor series (VTS) [Moreno et al., 1996]
  - Unscented transform [Julier and Uhlmann, 2004]

Noise compensation

- Aim: estimate $\boldsymbol{\mu}_y$ and $\boldsymbol{\Sigma}_y$ for each Gaussian component.
- Difficulty: $\mathbf{y} = f(\mathbf{x}, \mathbf{h}, \mathbf{n}, \boldsymbol{\alpha})$ is highly nonlinear, no analytic solution!
- Solution:
  - Vector Taylor series (VTS) [Moreno et al., 1996]
  - Unscented transform [Julier and Uhlmann, 2004]

Noise compensation

- Aim: estimate $\boldsymbol{\mu}_y$ and $\boldsymbol{\Sigma}_y$ for each Gaussian component.
- Difficulty: $\mathbf{y} = f(\mathbf{x}, \mathbf{h}, \mathbf{n}, \boldsymbol{\alpha})$ is highly nonlinear, no analytic solution!
- Solution:
  - Vector Taylor series (VTS) [Moreno et al., 1996]
  - Unscented transform [Julier and Uhlmann, 2004]

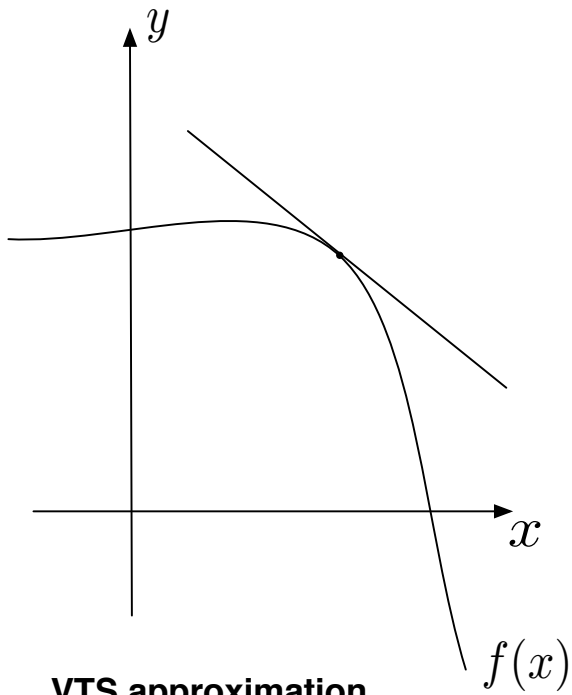
Noise compensation

- Aim: estimate $\boldsymbol{\mu}_y$ and $\boldsymbol{\Sigma}_y$ for each Gaussian component.
- Difficulty: $\mathbf{y} = f(\mathbf{x}, \mathbf{h}, \mathbf{n}, \boldsymbol{\alpha})$ is highly nonlinear, no analytic solution!
- Solution:
  - Vector Taylor series (VTS) [Moreno et al., 1996]
  - Unscented transform [Julier and Uhlmann, 2004]

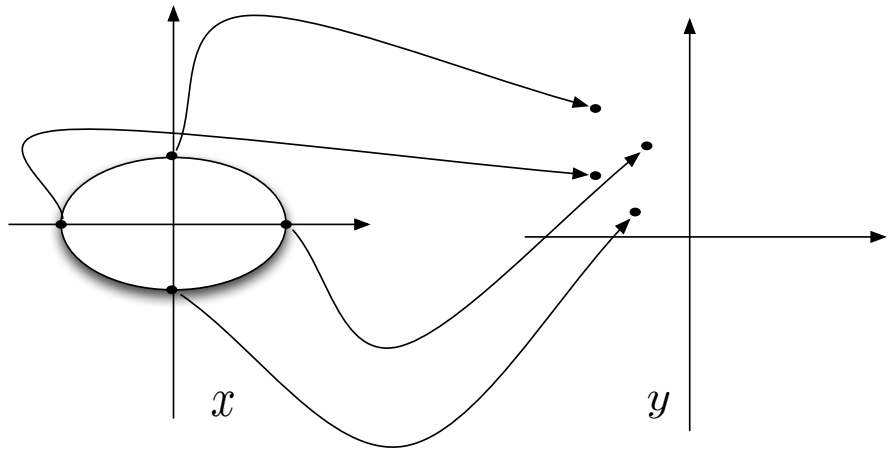
Noise compensation

- Aim: estimate $\boldsymbol{\mu}_y$ and $\boldsymbol{\Sigma}_y$ for each Gaussian component.
- Difficulty: $\mathbf{y} = f(\mathbf{x}, \mathbf{h}, \mathbf{n}, \boldsymbol{\alpha})$ is highly nonlinear, no analytic solution!
- Solution:
  - Vector Taylor series (VTS) [Moreno et al., 1996]
  - Unscented transform [Julier and Uhlmann, 2004]

Noise compensation



VTS approximation

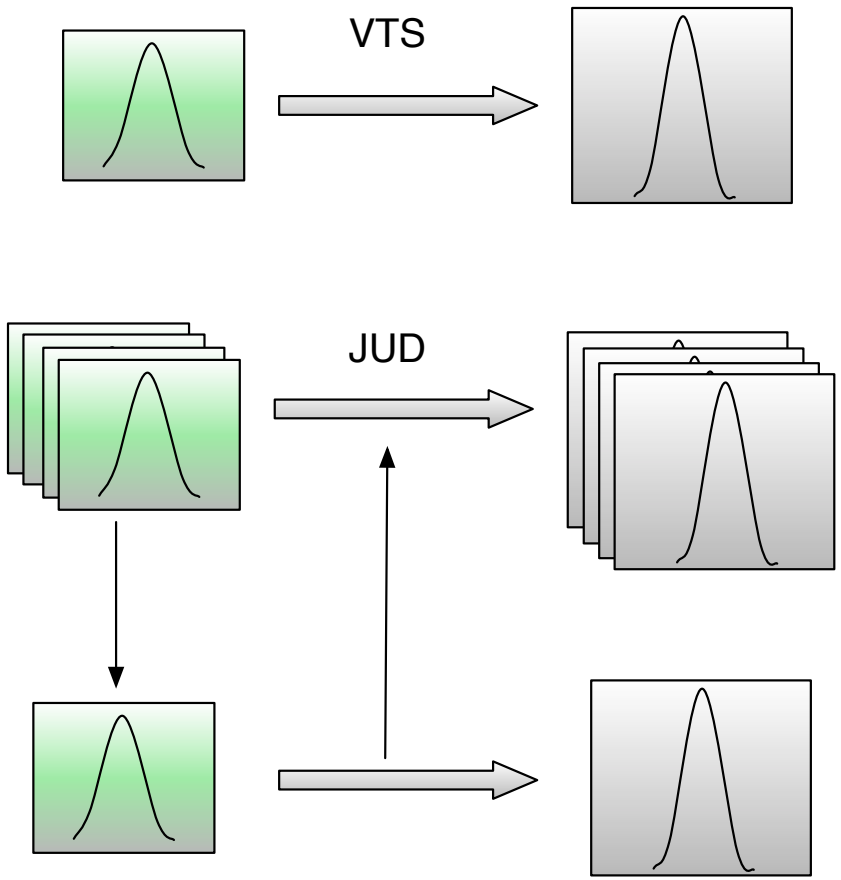


Unscented transform

- Cost: For VTS, real time factor > 100, memory $> 10G$ for (medium size) SGMM with 6.4M Gaussian

Noise compensation

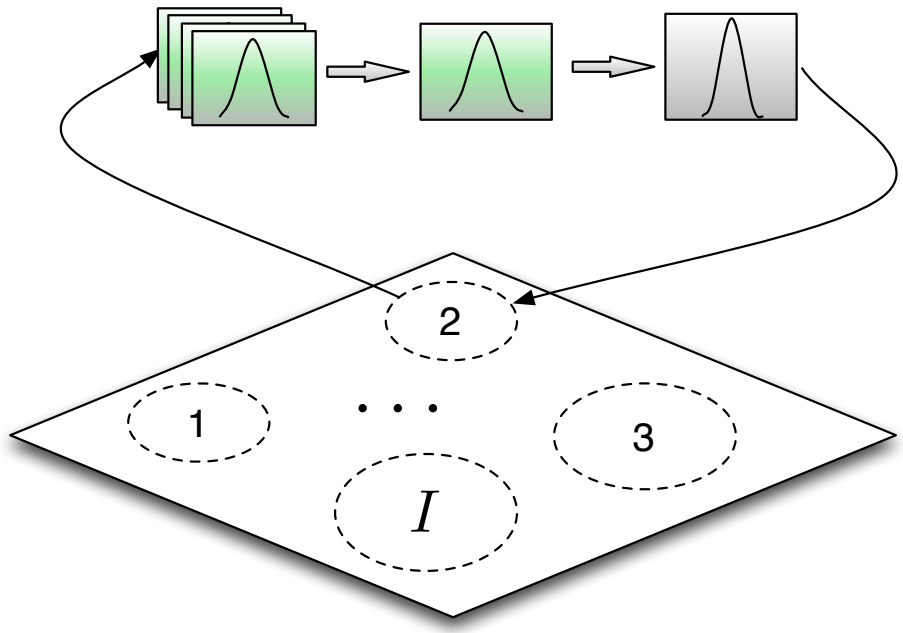
- Solution: Joint uncertainty decoding (JUD) [Liao and Gales, 2005]



VTS vs. JUD

Noise compensation

- Applying JUD to SGMM



- Cost: Real time factor $\sim$ 10 for SGMM with 6.4M Gaussians

Experiments

- Database
  - Aurora 4 dataset
  - Clean speech and noisy speech with SNR [5db - 15db]
  - Close-talking microphone and desk-mounted microphone
  - $\sim$ 15 hour training data
  - 330 testing utterances
- System configuration
  - 39dim MFCC
  - #triphone states: 3.1k (GMM) vs. 3.9k (SGMM)
  - #Gaussians: 50k (GMM) vs. 6.4M (SGMM)
  - #regression classes: 112 (GMM) vs. 400 (SGMM)

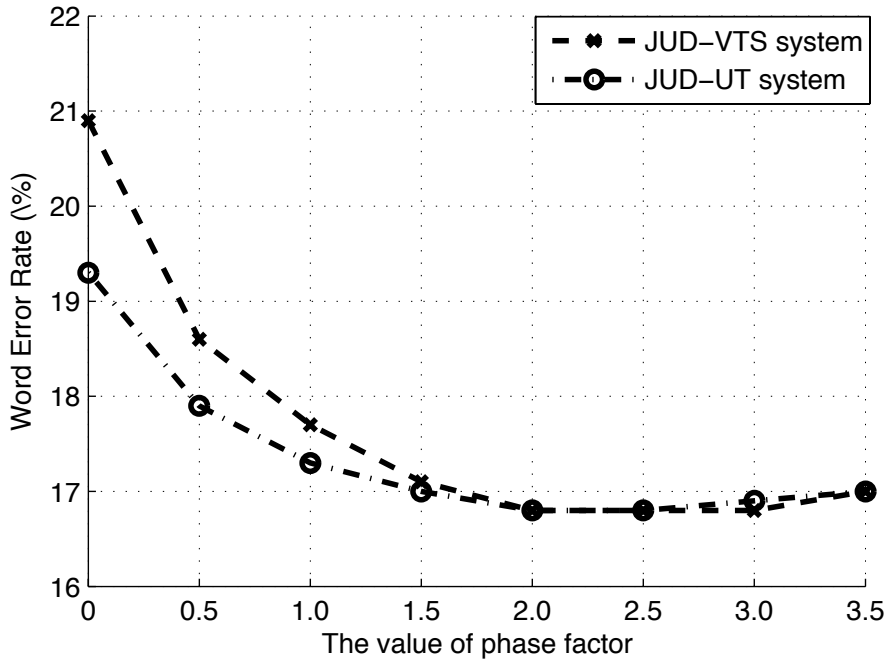
Noise compensation experiments

Baseline results without the phase term

System	WER
Baseline	**59.3**
JUD-VTS	**20.9**
+ UT	**20.0**
JUD-UT	**19.3**

Experiments

Results by tuning the value of phase factors.



- JUD/SGMM system achieved 16.8% WER on Aurora 4 database

Conclusion

- UT outperforms VTS when phase factor is not used
- Both of them achieve the same accuracy after tuning the phase term
- Noise compensation using JUD works well for SGMMs
- The phase term is particular effective for the noise compensation
- Future works will be on noise adaptive training, compensation in log-spectral domain.

Noise compensation

- With JUD, the marginal likelihood can be obtained as

$$p(\mathbf{y} \mid m) \approx |\mathbf{A}^{(r)}| \mathcal{N}\left(\mathbf{A}^{(r)}\mathbf{y} + \mathbf{b}^{(r)}; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m + \boldsymbol{\Sigma}_b^{(r)}\right). \quad (2)$$

- The transformation is done in the feature space, applied to each frame
- Computation is saved since that the #frame $\ll$ #Gaussians
- The transformation should be diagonalized in GMM systems, but not in SGMM system since we used full covariance matrix

Experiments

Table: GMM systems with $\boldsymbol{\alpha} = \mathbf{0}$.

Methods	Clean	Avg
Clean model	7.7	**59.3**
MTR model	12.7	**26.9**
VTS	7.3	**18.3**
JUD	7.0	**21.1**

Table: SGMM systems with $\boldsymbol{\alpha} = \mathbf{0}$.

Methods	Clean	Avg
Clean model	5.2	**59.9**
MTR model	6.8	**22.2**
JUD	5.3	**20.3**

Acero, A. (1990).
Acoustic and Enviromental Robustness in Automatic Speech Recognition.
PhD thesis, Carnegie Mellon University.

Deng, L., Droppo, J., and Acero, A. (2004).
Enhancement of log mel power spectra of speech using a phase-sensitive model of the acoustic environment and sequential estimation of the corrupting noise.
IEEE Transactions on Speech and Audio Processing, 12(2):133–143.

Julier, S. and Uhlmann, J. (2004).
Unscented filtering and nonlinear estimation.
Proceedings of the IEEE, 92(3):401–422.

Li, J., Deng, L., Yu, D., Gong, Y., and Acero, A. (2009).
A unified framework of HMM adaptation with joint compensation of additive and convolutive distortions.
Computer Speech & Language, 23(3):389–405.

Liao, H. and Gales, M. (2005).
Joint uncertainty decoding for noise robust speech recognition.
In *Proc. INTERSPEECH*. Citeseer.

Moreno, P., Raj, B., and Stern, R. (1996).
A vector Taylor series approach for environment-independent speech recognition.
In *Proc. ICASSP*, volume 2, pages 733–736. IEEE.

Povey, D., Burget, L., Agarwal, M., Akyazi, P., Kai, F., Ghoshal, A., Glembek, O., Goel, N., Karafiát, M., Rastrow, A., Rose, R., Schwarz, P., and Thomas, S. (2011).
The subspace Gaussian mixture model—A structured model for speech recognition.
Computer Speech & Language, 25(2):404–439.