Model-Based Approaches to Robust Speech Recognition

Mark Gales with Hank Liao, Rogier van Dalen, Chris Longworth
(work partly funded by Toshiba Research Europe Ltd)

11 June 2008

Cambridge University Engineering Department

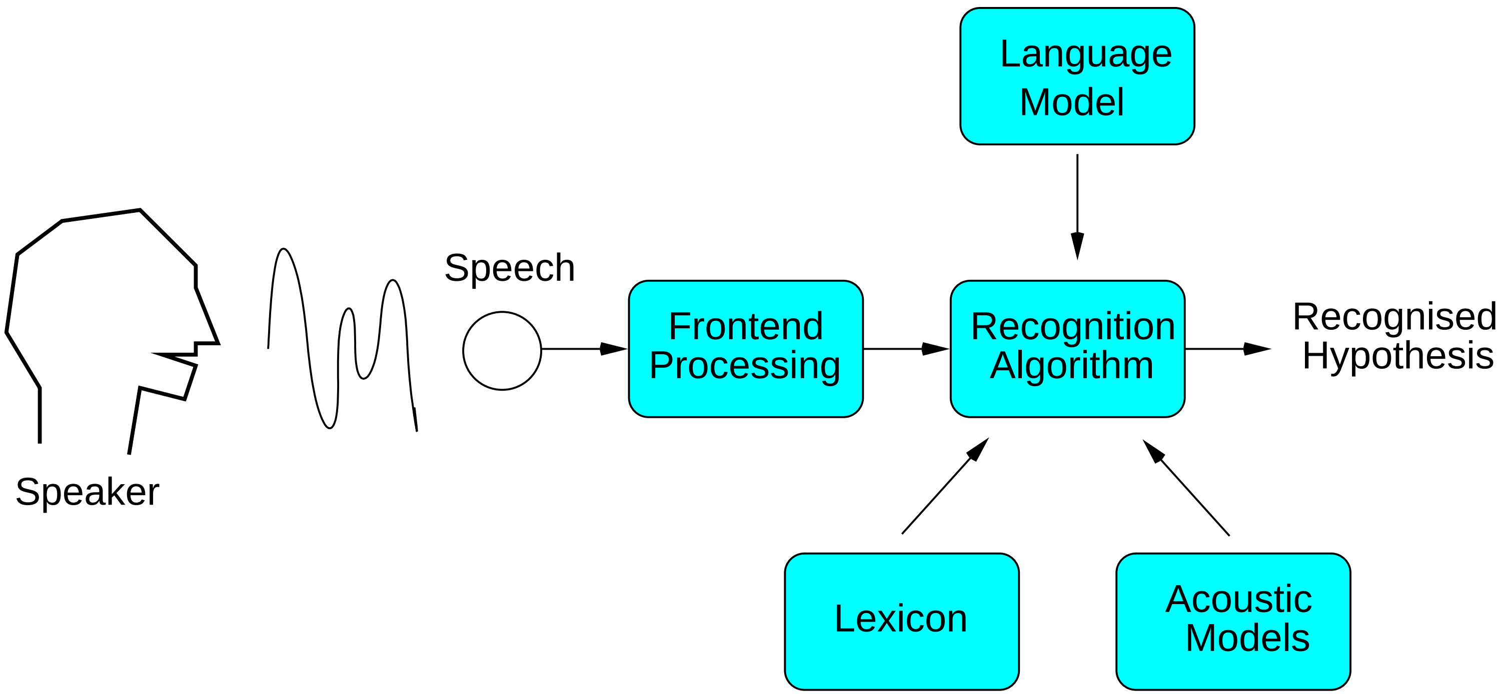
King's College London Seminar

Overview

- Speech recognition overview
- Noise robust speech recognition
  - impact of noise on acoustic features
  - "mismatch" functions
- Handling adverse environments
  - minimum mean-square error estimates
  - model-based compensation approaches
  - estimating the noise model parameters
- Model-based refinements
  - joint uncertainty decoding
  - covariance matrix modelling
  - generative kernels and SVMs

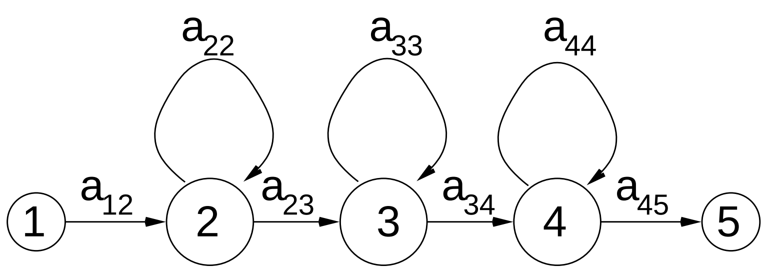
Example Application - In-Car Navigation

Speech Recognition Overview

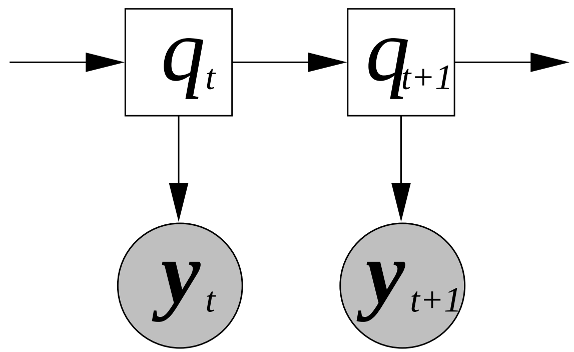


- Robust speech recognition (primarily) concerned with Acoustic models and Front-end processing
  - speech parameterised using continuous observations, MFCC [1] or PLP [2]
  - hidden Markov models used in the majority of speech recognition systems [3]

Hidden Markov Model - A Dynamic Bayesian Network



(a) Standard HMM phone topology



(b) HMM Dynamic Bayesian Network

- Notation for DBNs:

  circles - continuous variables   shaded - observed variables
  squares - discrete variables   non-shaded - unobserved variables

- Observations conditionally independent of other observations given state.
- States conditionally independent of other states given previous states.
- Poor model of the speech process - piecewise constant state-space.
  - but is the dominant acoustic model for speech recognition.

HMM Likelihood and Training

- HMM likelihood for sequence $\mathbf{Y} = \boldsymbol{y}_1, \ldots, \boldsymbol{y}_T$ is

$$p(\mathbf{Y}; \boldsymbol{\lambda}_{\mathrm{y}}) = \sum_{\mathbf{q} \in \mathbf{Q}} P(q_0) \prod_{t=1}^{T} P(q_t | q_{t-1}) p(\boldsymbol{y}_t | q_t)$$

- State output distributions modelled using Gaussian Mixture Models (GMMs)

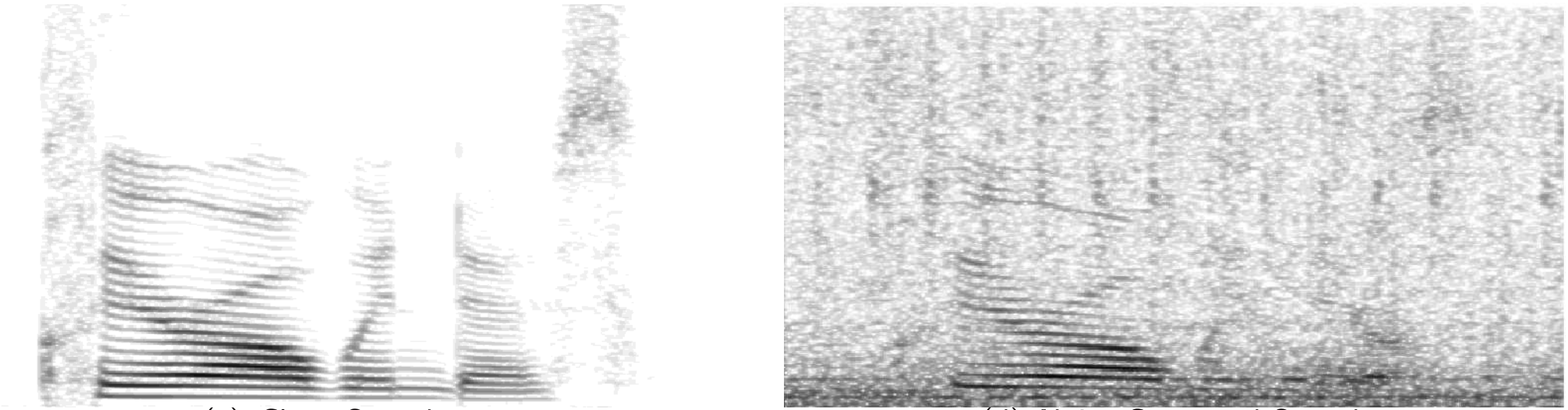
$$p(\boldsymbol{y}_t | j) = \sum_{m=1}^{M} c_{jm} \mathcal{N}(\boldsymbol{y}_t; \boldsymbol{\mu}^{(jm)}, \boldsymbol{\Sigma}^{(jm)})$$

- EM used to find the model parameters, mean estimated using

$$\hat{\boldsymbol{\mu}}_{\mathrm{y}}^{(m)} = \frac{\sum_{t=1}^{T} \gamma_{\mathrm{x}t}^{(m)} \boldsymbol{y}_t}{\sum_{t=1}^{T} \gamma_{\mathrm{y}t}^{(m)}}; \quad \gamma_{\mathrm{y}t}^{(m)} = P(q_t = m | \mathbf{Y}; \boldsymbol{\lambda}_{\mathrm{y}})$$

  - diagonal covariance matrices commonly used for memory/efficient reasons

Noise Robust Speech Recognition

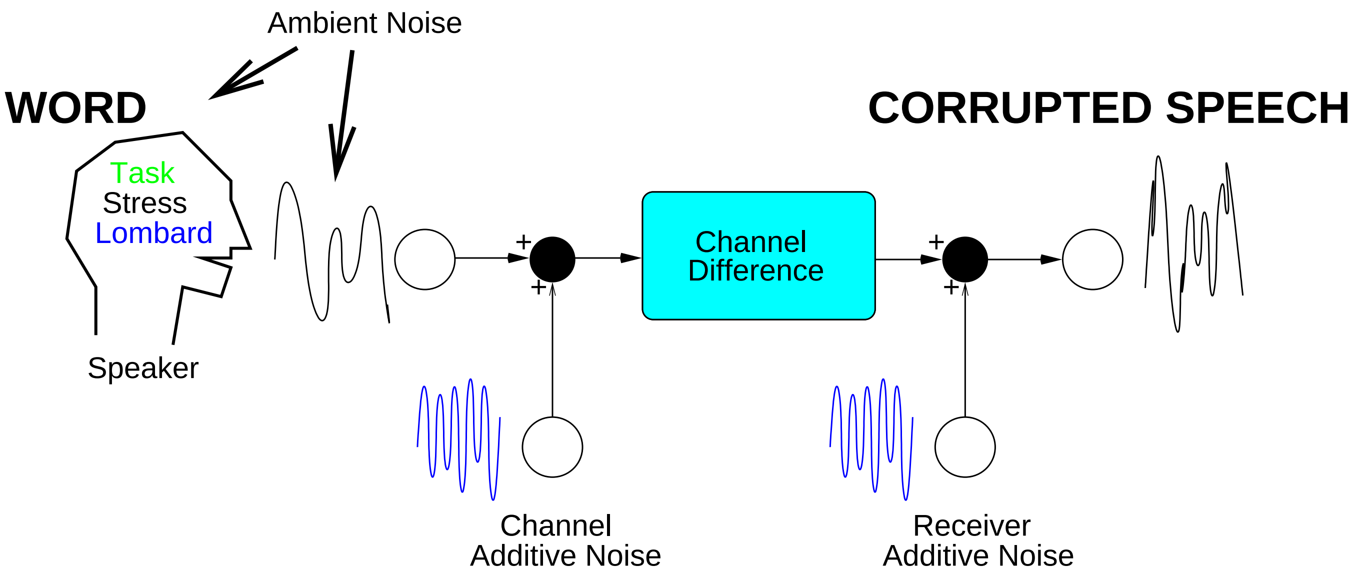


(c) Clean Speech

(d) Noise Corrupted Speech

- Background noise (and channel distortion) can seriously affect the signal
  - must be handled to enable ASR systems to work in e.g. in-car applications
- Need to quantify the impact that "noise" has on "clean" speech

General Environment Model



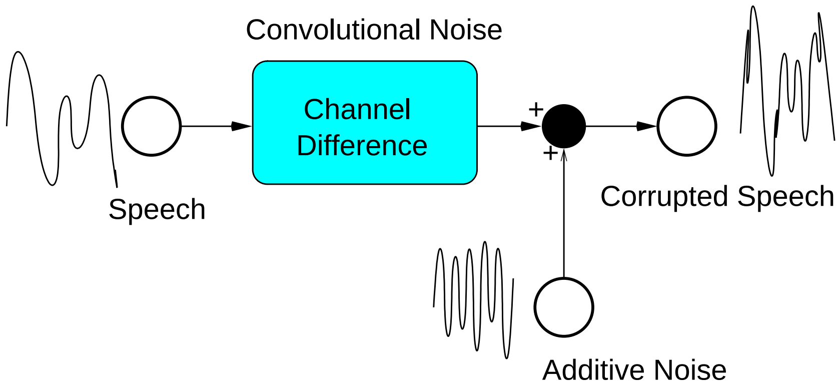
- The noise-corrupted speech, $y(t)$, and the noise-free speech , $x(t)$, related by

$$y(t) = \left[\left\{ \left(\left[x(t) \Big|^{\texttt{Stress}}_{\texttt{Lombard}} \right]_{n_1(t)} + n_1(t) \right) * h_{\texttt{mike}}(t) + n_2(t) \right\} * h_{\texttt{chan}}(t) \right] + n_3(t)$$

  – stress/Lombard not considered in this talk

"Simplified" Acoustic Environment

- A simplified model of the effects of noise is often used



- Ignore effects of stress:
- Group noise sources

$$y(t) = x(t) * h(t) + n(t)$$

- Squared magnitude of the Fourier Transform of signal

$$Y(f)Y^*(f) = |H(f)X(f)|^2 + |N(f)|^2 + 2|N(f)||H(f)X(f)|\cos(\theta)$$

θ is the angle between the vectors $N(f)$ and $H(f)X(f)$.

- Average (over Mel bins), assume speech and noise independent and $\log()$ [4]

$$\boldsymbol{y}_t^{\mathsf{l}} = \log\left(\exp\left(\boldsymbol{x}_t^{\mathsf{l}} + \boldsymbol{h}^{\mathsf{l}}\right) + \exp\left(\boldsymbol{n}_t^{\mathsf{l}}\right)\right)$$

Corrupted Speech Features

- Speech data is normally parameterised in the Cepstral domain, thus

$$\boldsymbol{y}_t^{\mathrm{s}} = \mathbf{C}\log\left(\exp(\mathbf{C}^{\text{-}1}\boldsymbol{x}_t^{\mathrm{s}} + \mathbf{C}^{\text{-}1}\boldsymbol{h}^{\mathrm{s}}) + \exp(\mathbf{C}^{\text{-}1}\boldsymbol{n}_t^{\mathrm{s}})\right) = \boldsymbol{x}_t^{\mathrm{s}} + \boldsymbol{h}^{\mathrm{s}} + f(\boldsymbol{x}_t^{\mathrm{s}}, \boldsymbol{n}_t^{\mathrm{s}}, \boldsymbol{h}^{\mathrm{s}})$$

  $\mathbf{C}$ is the DCT

  - non-linear relationship between the clean speech, noise and corrupted speech

- This has assumed sufficient smoothing to remove all “cross” terms
  - some sites use interaction likelihoods or phase-sensitive functions [5, 6]
  - given $\boldsymbol{x}_t^{\mathrm{s}}, \boldsymbol{h}^{\mathrm{s}}$ and $\boldsymbol{n}_t^{\mathrm{s}}$ there is a distribution

$$\boldsymbol{y}_t^{\mathrm{s}} \sim \mathcal{N}\left(\boldsymbol{x}_t^{\mathrm{s}} + \boldsymbol{h}_t^{\mathrm{s}} + f(\boldsymbol{x}_t^{\mathrm{s}}, \boldsymbol{n}_t^{\mathrm{s}}, \boldsymbol{h}^{\mathrm{s}}), \boldsymbol{\Phi}\right)$$

Delta and Delta-Delta Parameters

- Feature vector modified to 'reduce' HMM conditional independence assumptions
  - standard to add delta and delta-delta [7] parameters

$$\boldsymbol{y}_t = \begin{bmatrix} \boldsymbol{y}^{\mathrm{s}}_t \\ \boldsymbol{\Delta y}^{\mathrm{s}}_t \\ \boldsymbol{\Delta}^2\boldsymbol{y}^{\mathrm{s}}_t \end{bmatrix}; \quad \boldsymbol{\Delta y}^{\mathrm{s}}_t = \frac{\sum_{i=1}^{n} w_i \left(\boldsymbol{y}^{\mathrm{s}}_{t+i} - \boldsymbol{y}^{\mathrm{s}}_{t-i}\right)}{\sum_{i=1}^{n} w_i^2}$$

- Two versions used to represent the impact of noise on these [8]

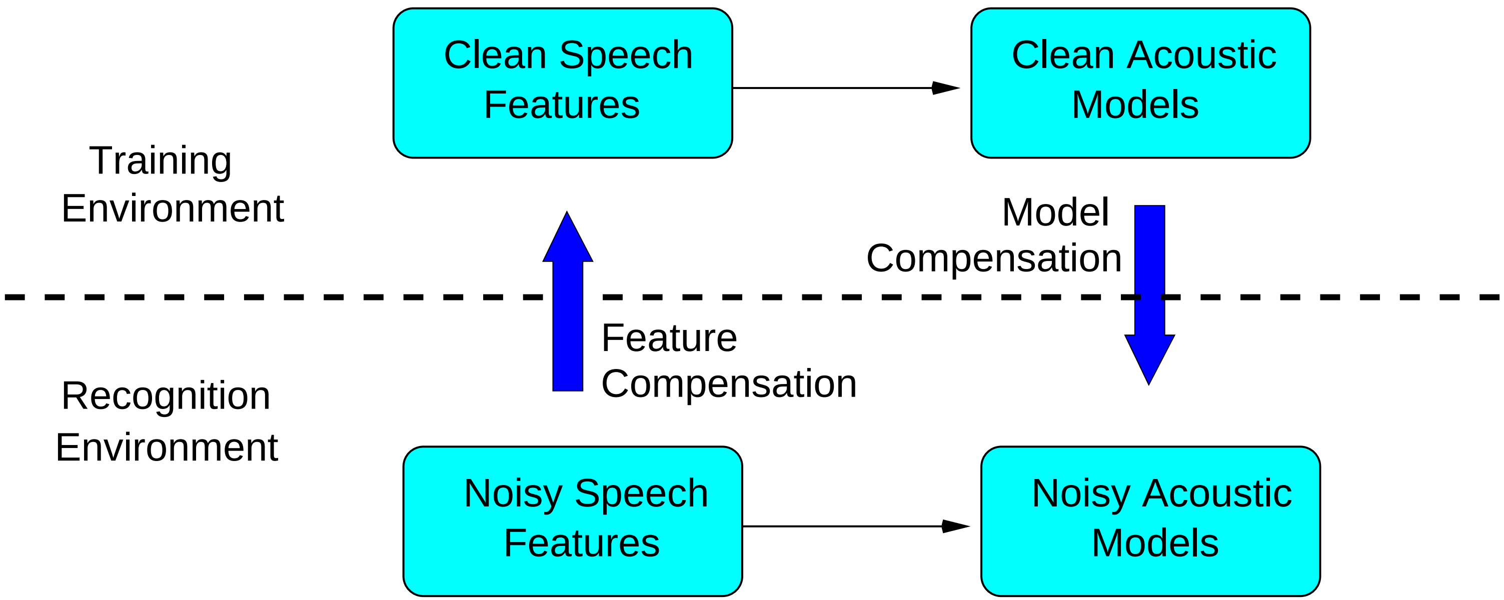
$$\boldsymbol{\Delta y}^{\mathrm{s}}_t \approx \frac{\partial \boldsymbol{y}^{\mathrm{s}}_t}{\partial t} \quad \text{OR} \quad \boldsymbol{\Delta y}^{\mathrm{s}}_t = \mathbf{D}\begin{bmatrix} \boldsymbol{y}^{\mathrm{s}}_{t-1} \\ \boldsymbol{y}^{\mathrm{s}}_t \\ \boldsymbol{y}^{\mathrm{s}}_{t+1} \end{bmatrix}$$

  - the second is more accurate, but more statistics required to be stored
  - need to compensate all model parameters for best performance

Dealing with Adverse Environments

- Single-microphone techniques may be split into
  - inherently robust speech parameterisation - no modifications to the system.
  - clean speech estimation - alters the front-end processing scheme.
  - acoustic model compensation so that they are representative of speech in the new acoustic environment.
- Multiple-microphones - microphone arrays may be used
  - increase SNR by reducing the beam-width of the effective microphone.
  - additional/specialised hardware required
- If something is known about the possible test acoustic environment
  - multi-style (multi-environment) training may be used
  - "clean" model trained under a variety of conditions
  - also helps general robustness
- Talk concentrates on single-microphone approaches.

Noise Compensation Approaches



- Two main approaches:
  - feature compensation: "clean" the noisy features
  - model compensation: "corrupt" the clean models
- Some schemes, e.g. feature uncertainty, share properties of both.

Minimum Mean-Square Error Estimates

- Estimate the clean speech $\hat{\boldsymbol{x}}_t$ given the corrupted speech $\boldsymbol{y}_t$
  - to handle non-linearity partition space using an R-component GMM, then

$$\hat{\boldsymbol{x}}_t = \mathcal{E}\{\boldsymbol{x}_t|\boldsymbol{y}_t\} = \sum_{r=1}^{R} P(r|\boldsymbol{y}_t)\mathcal{E}\{\boldsymbol{x}_t|\boldsymbol{y}_t, r\}$$

- Model the joint-distribution for each component, then [9]

$$\left.\left[\begin{array}{c} \boldsymbol{y}_t \\ \boldsymbol{x}_t \end{array}\right]\right| r \sim \mathcal{N}\left(\left[\begin{array}{c} \boldsymbol{\mu}_{\rm y}^{(r)} \\ \boldsymbol{\mu}_{\rm x}^{(r)} \end{array}\right], \left[\begin{array}{cc} \boldsymbol{\Sigma}_{\rm yy}^{(r)} & \boldsymbol{\Sigma}_{\rm yx}^{(r)} \\ \boldsymbol{\Sigma}_{\rm xy}^{(r)} & \boldsymbol{\Sigma}_{\rm xx}^{(r)} \end{array}\right]\right)$$

$$\mathcal{E}\{\boldsymbol{x}_t|\boldsymbol{y}_t, r\} = \boldsymbol{\mu}_{\rm x}^{(r)} + \boldsymbol{\Sigma}_{\rm xy}^{(r)}\boldsymbol{\Sigma}_{\rm yy}^{(r)\text{-}1}(\boldsymbol{y}_t - \boldsymbol{\mu}_{\rm y}^{(r)}) = \mathbf{A}^{(r)}\boldsymbol{y}_t + \mathbf{b}^{(r)}$$

  - joint distribution estimated using stereo data
    can be estimated using model-based compensation schemes
  - various forms/variants possible: SPLICE [10], POF[11]

General Model Adaptation

- A standard scheme for speaker/environment adaptation is linear transforms

  Various forms used [12, 13]:

  - MLLR Mean: $\boldsymbol{\mu}_{\rm y}^{(m)} = \mathbf{A}\boldsymbol{\mu}_{\rm x}^{(m)} + \mathbf{b}$
  - MLLR Variance: $\boldsymbol{\Sigma}_{\rm y}^{(m)} = \mathbf{A}\boldsymbol{\Sigma}_{\rm x}^{(m)}\mathbf{A}^{\mathsf{T}}$
  - CMLLR: $\boldsymbol{y}_t = \mathbf{A}\boldsymbol{x}_t + \mathbf{b}$ (MLLR mean/variance transforms same)

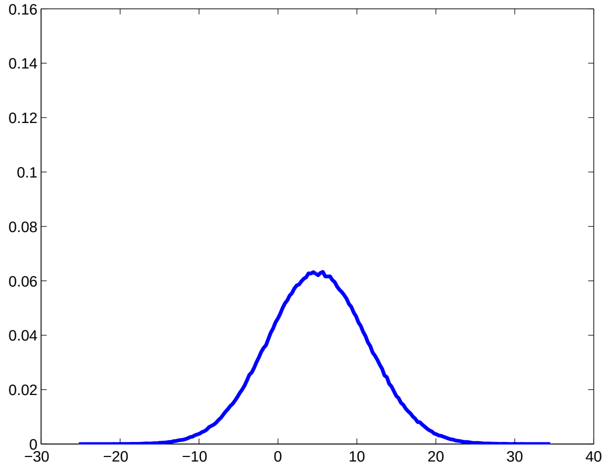
- Transforms usually estimated using maximum likelihood and EM

$$\left\{\hat{\mathbf{A}}, \hat{\mathbf{b}}\right\} = \underset{\mathbf{A},\mathbf{b}}{\operatorname{argmax}} \left\{p(\mathbf{Y}|\mathbf{A}, \mathbf{b}; \boldsymbol{\lambda}_{\rm x})\right\}$$

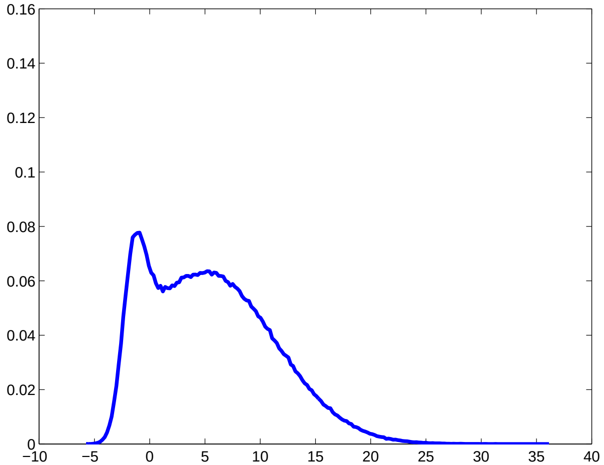
- Problems include:

  - large numbers of model parameters need to be estimated ($\mathbf{A}$ usually full)
  - for unsupervised adaptation require a hypothesis $\mathcal{H}$ for utterance $\mathbf{Y}$.

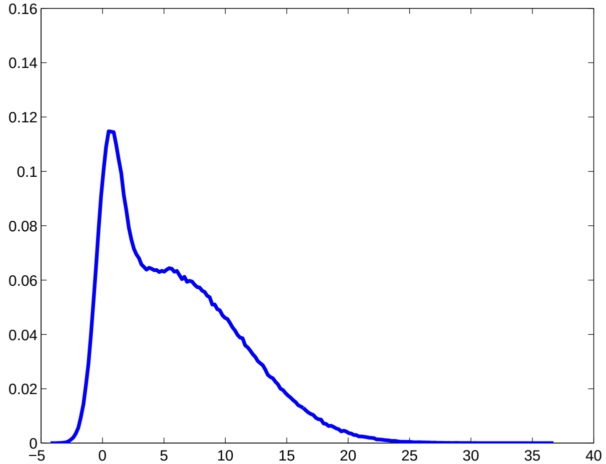
Effects of Additive Noise



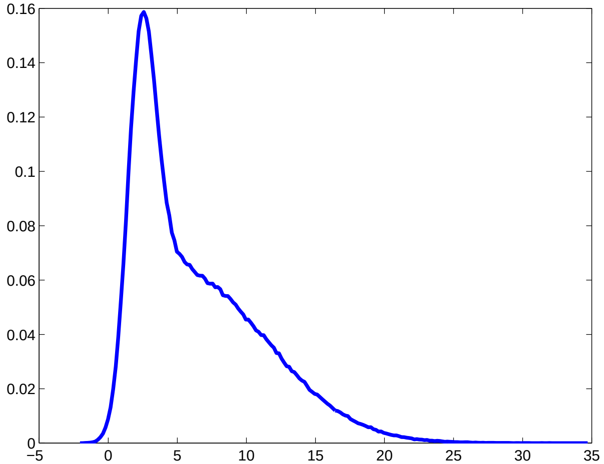
(e) Clean Speech



(f) Noise mean=-2



(g) Noise mean=0



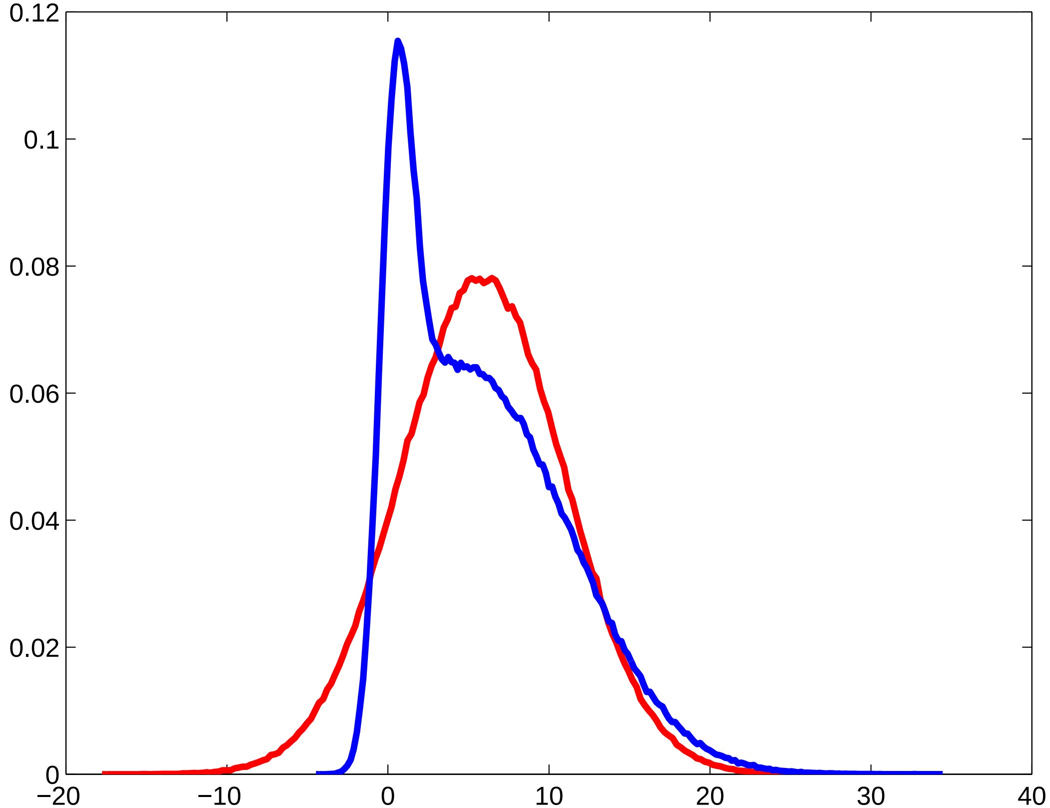
(h) Noise mean=2

Model-Based Adaptation using Stereo Data

- The simplest model-based compensation scheme is to make use of stereo/noise corrupted data
  - $\mathbf{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T\}$: clean speech samples
  - $\mathbf{Y} = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_T\}$: corrupted speech samples
- For stereo data $\boldsymbol{y}_t$ is the noise corrupted version of $\boldsymbol{x}_t$
- Two choices for training systems
  - train in the standard fashion on the noise corrupted data
  - use single-pass retraining (SPR) [14]

$$\boldsymbol{\mu}_{\mathrm{y}}^{(m)} = \frac{\sum_{t=1}^{T} \gamma_{\mathrm{x}t}^{(m)} \boldsymbol{y}_t}{\sum_{t=1}^{T} \gamma_{\mathrm{x}t}^{(m)}}; \quad \gamma_{\mathrm{x}t}^{(m)} = P(q_t = m | \mathbf{X}; \boldsymbol{\lambda}_{\mathrm{x}})$$

Single-Gaussian Approximation



- Single-pass retraining uses complete data-set from the clean system ($\gamma_{\mathtt{x}t}^{(m)}$)
  - approximates corrupted distribution using a single Gaussian

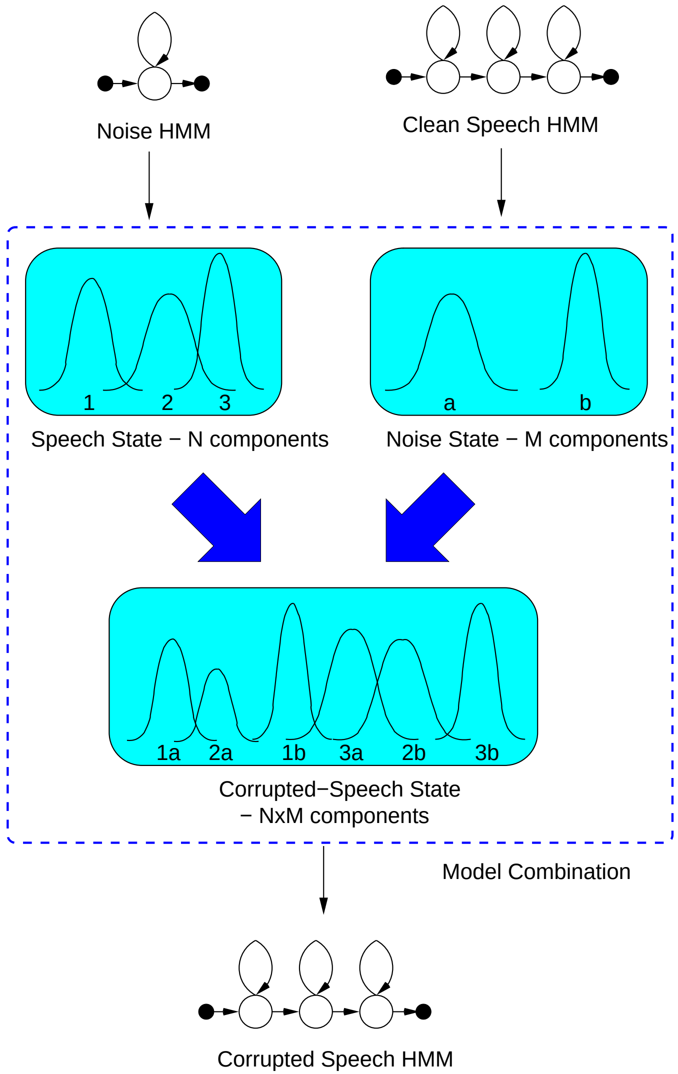
Model-Based Compensation

- SPR is "accurate" but slow
  - need to have all training data available and corrupt it with noise
- Model-based compensation approximates SPR [14]

$$\boldsymbol{\mu}_{\mathrm{y}}^{(m)} = \mathcal{E}\{\boldsymbol{y}|m\}; \qquad \boldsymbol{\Sigma}_{\mathrm{y}}^{(m)} = \mathsf{diag}\left(\mathcal{E}\{\boldsymbol{y}\boldsymbol{y}^{\mathsf{T}}|m\} - \boldsymbol{\mu}_{\mathrm{y}}^{(m)}\boldsymbol{\mu}_{\mathrm{y}}^{(m)\mathsf{T}}\right)$$

- Due to non-linearities no closed form solution - approximations required
  - Monte-Carlo-style: generate "speech" and "noise" observations and combine
  - Log-Add: only transform the mean
  - Log-Normal: sum of two log-normal variables approximately log-normal
  - Vector Taylor series: first or higher order expansions used

Model-Based Compensation Procedure



- Process for log-add approximation [14] is:

1. Map to log-Spectral domain

$$\boldsymbol{\mu}^{\mathsf{l}} = \mathbf{C}^{-1}\boldsymbol{\mu}^{\mathsf{s}}; \quad \boldsymbol{\Sigma}^{\mathsf{l}} = \mathbf{C}^{-1}\boldsymbol{\Sigma}^{\mathsf{s}}(\mathbf{C}^{-1})^{\mathsf{T}}$$

2. Map to linear spectral domain

$$\begin{aligned} \mu_i^{\mathsf{f}} &= \exp\{\mu_i^{\mathsf{l}} + \sigma_{ii}^{\mathsf{l}}/2\} \\ \sigma_{ij}^{\mathsf{f}} &= \mu_i^{\mathsf{f}}\mu_j^{\mathsf{f}}(\exp\{\sigma_{ij}^{\mathsf{l}}\} - 1) \end{aligned}$$

3. Combine speech and noise models

$$\boldsymbol{\mu}_{\mathsf{y}}^{\mathsf{f}} = \boldsymbol{\mu}_{\mathsf{x}}^{\mathsf{f}} + \boldsymbol{\mu}_{\mathsf{n}}^{\mathsf{f}}; \quad \boldsymbol{\Sigma}_{\mathsf{y}}^{\mathsf{f}} = \boldsymbol{\Sigma}_{\mathsf{x}}^{\mathsf{f}} + \boldsymbol{\Sigma}_{\mathsf{n}}^{\mathsf{f}}$$

4. Map back to Cepstral domain

Vector Taylor Series

- Vector Taylor Series (VTS) one popular approximation [15, 16]
  - Taylor series expansion about "current" parameter values
  - for these expression ignore impact of convolutional distortion
  - mismatch function approximated using first order series

$$\boldsymbol{y}_t^{\mathrm{s}} \approx \boldsymbol{\mu}_{\mathrm{x}}^{\mathrm{s}} + f(\boldsymbol{\mu}_{\mathrm{x}}^{\mathrm{s}}, \boldsymbol{\mu}_{\mathrm{n}}^{\mathrm{s}}) + \boldsymbol{\nabla}_{\mathrm{x}} f(\boldsymbol{x}, \boldsymbol{n})|_{\mu_{\mathrm{x}}^{\mathrm{s}}, \mu_{\mathrm{n}}^{\mathrm{s}}}(\boldsymbol{x}_t^{\mathrm{s}} - \boldsymbol{\mu}_{\mathrm{x}}^{\mathrm{s}}) + \boldsymbol{\nabla}_{\mathrm{n}} f(\boldsymbol{x}, \boldsymbol{n})|_{\mu_{\mathrm{x}}^{\mathrm{s}}, \mu_{\mathrm{n}}^{\mathrm{s}}}(\boldsymbol{n}_t^{\mathrm{s}} - \boldsymbol{\mu}_{\mathrm{n}}^{\mathrm{s}})$$

where $f(\boldsymbol{x}, \boldsymbol{n})$ is the mismatch function from previous slide (ignoring $\boldsymbol{h}^{\mathrm{s}}$)

- Gives simple approach to estimating noise parameters

$$\boldsymbol{\mu}_{\mathrm{y}}^{(m)\mathrm{s}} = \mathcal{E}\{\boldsymbol{y}_t^{\mathrm{s}}|m\} \approx \boldsymbol{\mu}_{\mathrm{x}}^{(m)\mathrm{s}} + f(\boldsymbol{\mu}_{\mathrm{x}}^{(m)\mathrm{s}}, \boldsymbol{\mu}_{\mathrm{n}}^{\mathrm{s}})$$

$$\boldsymbol{\Sigma}_{\mathrm{y}}^{(m)\mathrm{s}} \approx \mathbf{A}\boldsymbol{\Sigma}_{\mathrm{x}}^{(m)\mathrm{s}}\mathbf{A}^{\mathsf{T}} + (\mathbf{I} - \mathbf{A})\boldsymbol{\Sigma}_{\mathrm{n}}^{(m)\mathrm{s}}(\mathbf{I} - \mathbf{A})^{\mathsf{T}}; \quad \mathbf{A} = \frac{\partial \boldsymbol{y}^{\mathrm{s}}}{\partial \boldsymbol{x}^{\mathrm{s}}}$$

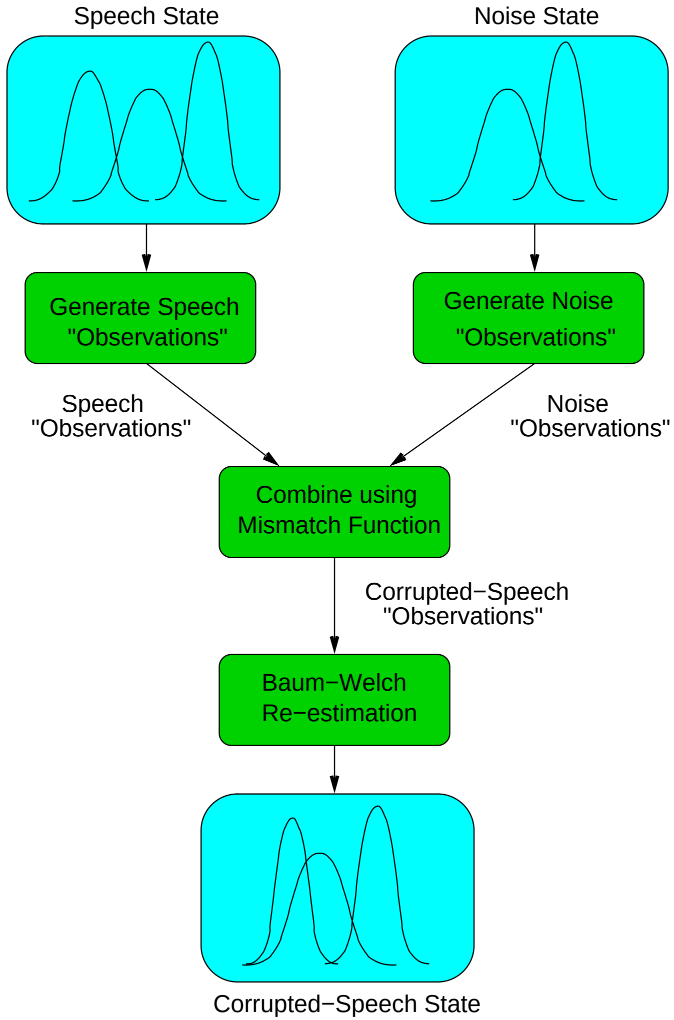
Noise Parameter Estimation

- In practice the noise model parameters, $\boldsymbol{\mu}_{\texttt{n}}, \boldsymbol{\mu}_{\texttt{h}}, \boldsymbol{\Sigma}_{\texttt{n}}$, are not known
  - need to be estimated from test data
  - simplest approach - use VAD and start/end frames to estimate noise
- Also possible to use ML estimation [15, 17]

$$\left\{\hat{\boldsymbol{\mu}}_{\texttt{n}}, \hat{\boldsymbol{\mu}}_{\texttt{h}}, \hat{\boldsymbol{\Sigma}}_{\texttt{n}}\right\} = \underset{\boldsymbol{\mu}_{\texttt{n}}, \boldsymbol{\mu}_{\texttt{h}}, \boldsymbol{\Sigma}_{\texttt{n}}}{\operatorname{argmax}} \left\{p(\mathbf{Y} | \boldsymbol{\mu}_{\texttt{n}}, \boldsymbol{\mu}_{\texttt{h}}, \boldsymbol{\Sigma}_{\texttt{n}}; \boldsymbol{\lambda}_{\texttt{x}})\right\}$$

- VTS approximation yields simple approach to find $\boldsymbol{\mu}_{\texttt{n}}, \boldsymbol{\mu}_{\texttt{h}}$
  - first/second-order approaches to find $\boldsymbol{\Sigma}_{\texttt{n}}$
  - simple statistics for auxiliary function

Iterative Approaches

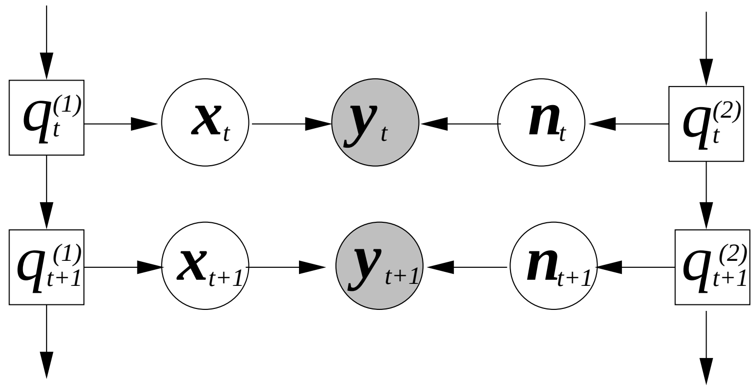


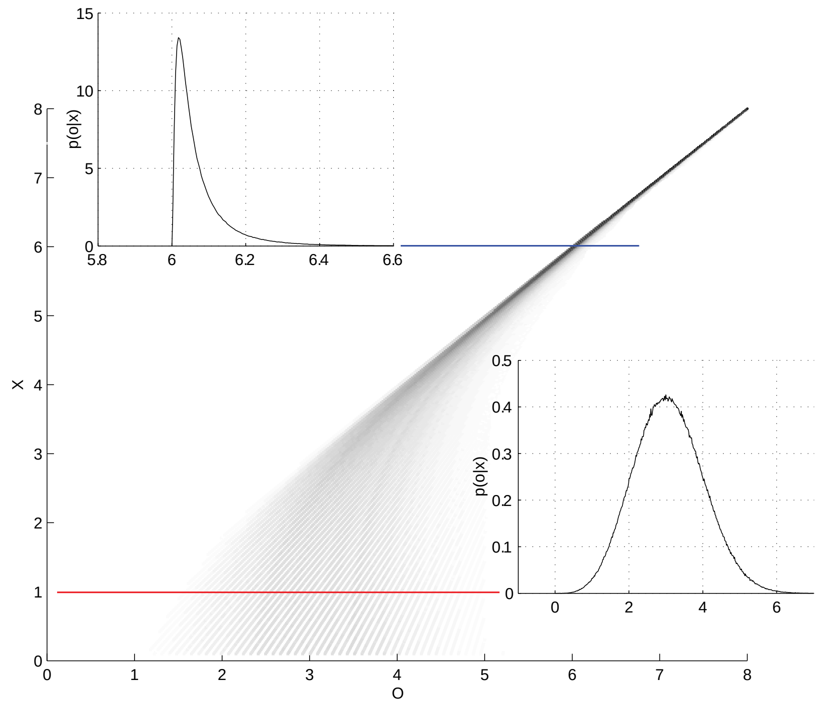
- Previous approaches use single-Gaussian approximation
  - iterative approaches relax this
  - two approaches in literature
- Algonquin: 'best' Gaussian approximation[5]
  - approximation varies according to $\boldsymbol{y}_t$
  - expensive as changes each frame
- DPMC: use non-Gaussian approximation [14]
  - Monte-Carlo-style with GMMs/state
  - expensive model-compensation scheme

Extensions to Model-Based Approaches

- Joint Uncertainty Decoding:
  - attempts to speed up model compensation process
- Predictive Linear Transforms:
  - efficiently handles changes in the feature-vector correlations
- SVM-Based Robust ASR:
  - combines model-based compensation with a discriminative classifier (SVM)

Uncertainty Decoding





$$p(\boldsymbol{y}_t) = \int p(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{n}_t)p(\boldsymbol{x}_t)p(\boldsymbol{n}_t)$$

- All the model-based approaches are computationally expensive
  - scales linearly with # components (100K+ for LVCSR systems)
- Need to model the conditional distribution $p(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{n}_t)$ [18, 5, 17]
  - select form to allow efficient compensation/decoding (if possible)

Joint Uncertainty Decoding

- Rather than model $p(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{n}_t)$ use [17]

$$p(\boldsymbol{y}_t|\boldsymbol{x}_t) = \int p(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{n}_t)p(\boldsymbol{n}_t)d\boldsymbol{n}_t$$

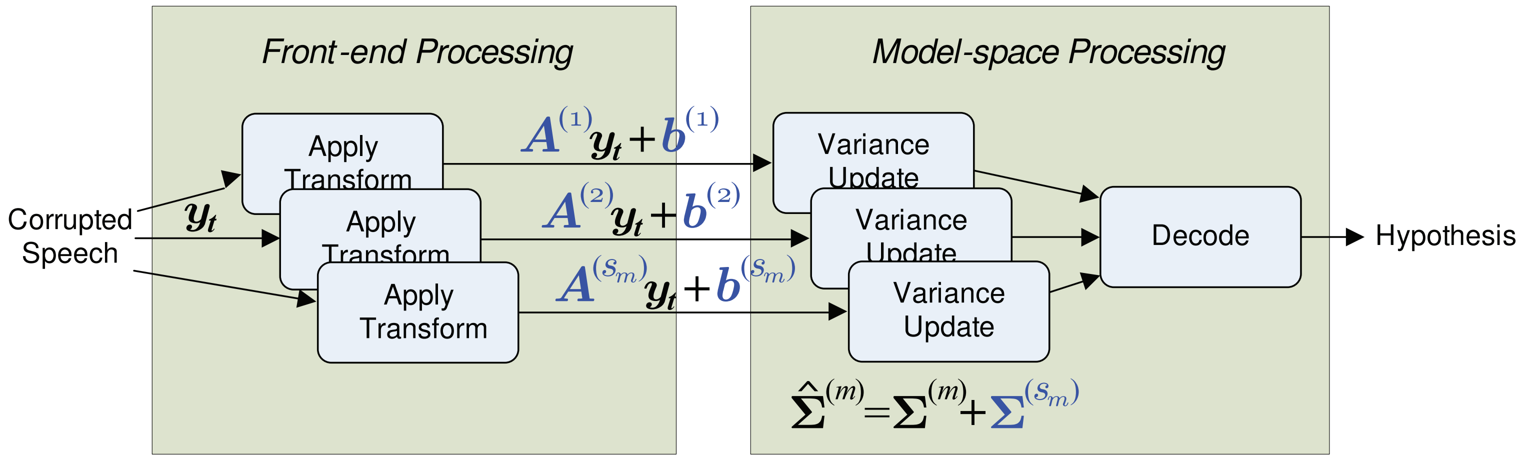
- Simplest approach is to assume $\boldsymbol{y}_t$ and $\boldsymbol{x}_t$ jointly Gaussian (again)
  - to handle changes with acoustic-space make dependent on r
  - simple to derive conditional distribution $p(\boldsymbol{y}_t|\boldsymbol{x}_t, r)$
  - contrast to MMSE where $p(\boldsymbol{x}_t|\boldsymbol{y}_t, r)$ modelled
  - joint distribution estimated using VTS/PMC (stereo data can also be used)
- Product of Gaussians is an un-normalised Gaussian, so

$$p(\boldsymbol{y}_t|m, r) = |\mathbf{A}^{(r)}|\mathcal{N}(\mathbf{A}^{(r)}\boldsymbol{y}_t + \mathbf{b}^{(r)}; \boldsymbol{\mu}^{(m)}, \boldsymbol{\Sigma}^{(m)} + \boldsymbol{\Sigma}_{\mathsf{b}}^{(r)})$$

  - r is normally determined by the component m [19]
  - contrast to MMSE where GMM built in acoustic space to determine r

JUD versus CMLLR

- For JUD compensation, PMC/VTS only required at regression class level
  - $\mathbf{A}^{(r)}, \mathbf{b}^{(r)}$ and $\boldsymbol{\Sigma}_{\mathrm{b}}^{(r)}$ functions of noise parameters $\boldsymbol{\mu}_{\mathrm{n}}, \boldsymbol{\mu}_{\mathrm{h}}, \boldsymbol{\Sigma}_{\mathrm{n}}$



- Similar to CMLLR however
  - JUD parameters estimated using noise models derived from data
  - CMLLR directly uses data to estimate parameters
  - JUD has a bias variance, found to be important for noise estimation

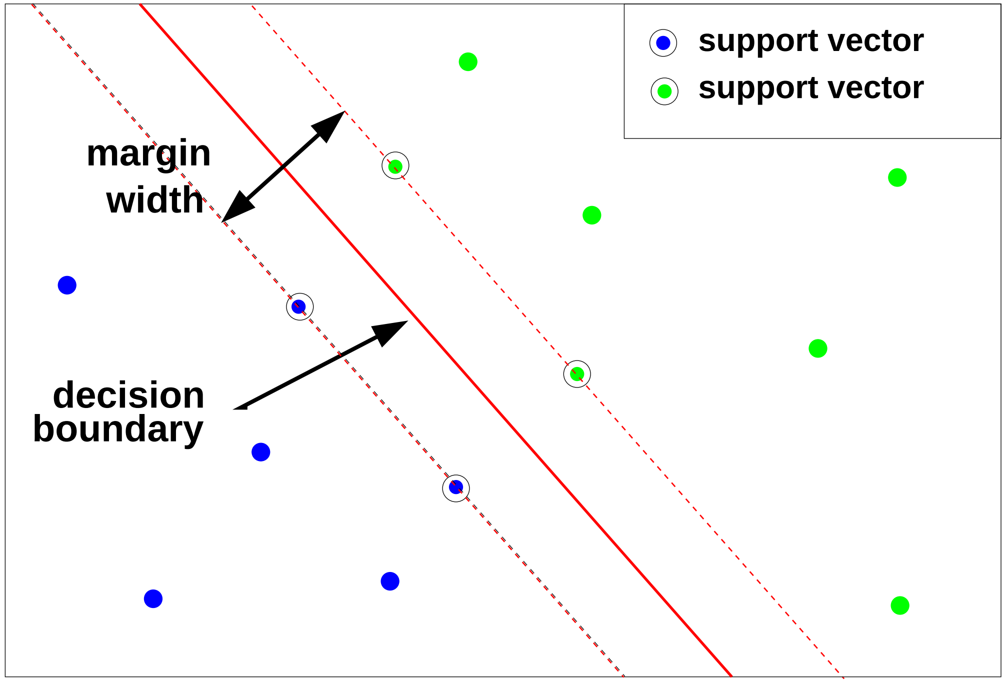
Full Covariance Matrix Modelling

- Background noise affects the correlation between elements of the feature-vector
  - normally diagonal covariance matrices used
  - useful to model correlation changes - use full $\mathbf{A}^{(r)}, \mathbf{b}^{(r)}$ and $\boldsymbol{\Sigma}_{\mathtt{b}}^{(r)}$
  - computationally expensive - full covariance decode $(\boldsymbol{\Sigma}^{(m)} + \boldsymbol{\Sigma}_{\mathtt{b}}^{(r)})$
- Standard schemes for efficient covariance/precision matrix modelling [3]
  - One example is semi-tied covariance matrices [20]

$$\left(\boldsymbol{\Sigma}^{(m)} + \boldsymbol{\Sigma}_{\mathtt{b}}^{(r)}\right)^{\text{-1}} = \mathbf{A}_{\mathtt{stc}}^{(r)\mathsf{T}} \boldsymbol{\Sigma}_{\mathtt{diag}}^{(m)\text{-1}} \mathbf{A}_{\mathtt{stc}}^{(r)}$$

  - Decoding efficient - $|\mathbf{A}_{\mathtt{stc}}^{(r)}| \mathcal{N}(\mathbf{A}_{\mathtt{stc}}^{(r)} \boldsymbol{y}_t; \boldsymbol{\mu}^{(m)}, \boldsymbol{\Sigma}_{\mathtt{diag}}^{(m)})$
- $\mathbf{A}_{\mathtt{stc}}^{(r)}$ can be found using statistics from JUD
  - a version of predictive linear transforms [21]

Support Vector Machines



- SVMs are a maximum margin, binary, classifier [22]:
  - related to minimising generalisation error;
  - unique solution (compare to neural networks);
  - may be kernelised - training/classification a function of dot-product $(\mathbf{x}_i.\mathbf{x}_j)$.
- Can be applied to speech - use a kernel to map variable data to a fixed length.

Generative Kernels

- Generative models, e.g. HMMs and GMMs, handle variable length data
  - view as "mapping" sequence to a single dimension (log-likelihood)

$$\phi\left(\mathbf{Y};\boldsymbol{\lambda}\right) = \frac{1}{T}\log\left(p(\mathbf{Y};\boldsymbol{\lambda})\right)$$

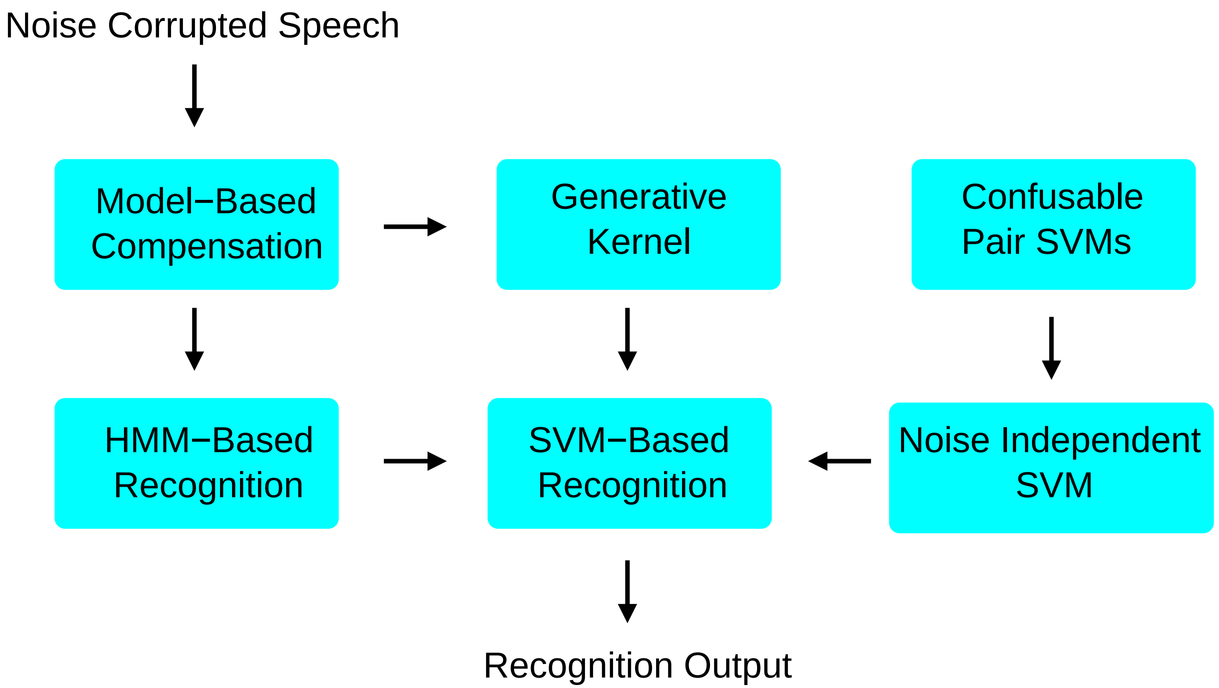
- Extend feature-space to a high dimension:
  - add derivatives with respect to the model parameters
  - example is a log-likelihood ratio plus first derivative score-space [23]:

$$\boldsymbol{\phi}(\mathbf{Y};\boldsymbol{\lambda}) = \frac{1}{T}\left[\begin{array}{c} \log\left(p(\mathbf{Y};\boldsymbol{\lambda}^{(1)})\right) - \log\left(p(\mathbf{Y};\boldsymbol{\lambda}^{(2)})\right) \\ \boldsymbol{\nabla}_{\lambda^{(1)}}\log\left(p(\mathbf{Y};\boldsymbol{\lambda}^{(1)})\right) \\ -\boldsymbol{\nabla}_{\lambda^{(2)}}\log\left(p(\mathbf{Y};\boldsymbol{\lambda}^{(2)})\right) \end{array}\right]$$

  - Related to the Fisher kernel [24]

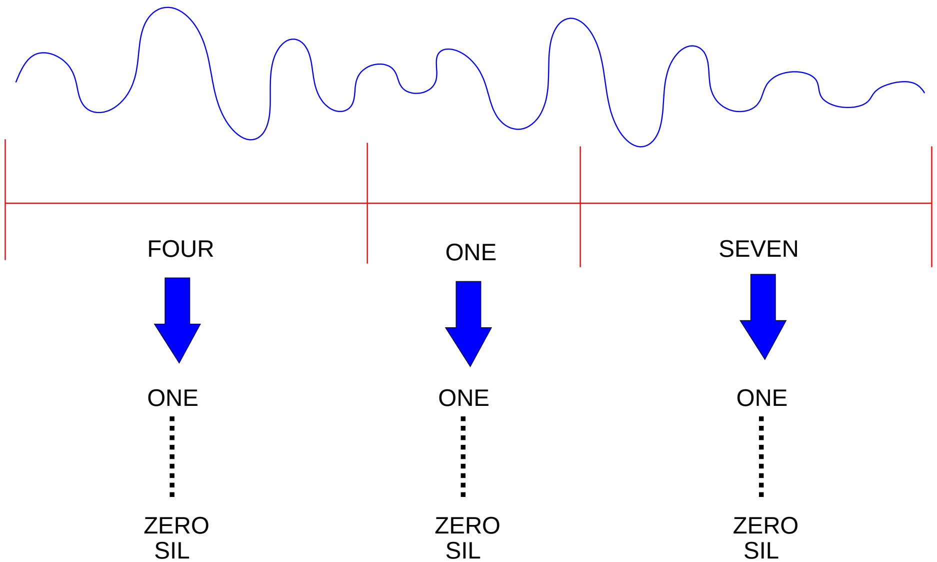
SVMs for Noise Robust ASR

- Difficult to adapt a SVM classifier to a noise condition [25]
  - adapt generative kernel model to the noise condition
  - leave the SVM classifier the same for all conditions



- How to handle large number of possible classes even for simple digit strings?

Handling Continuous Digit Strings



- Using HMM-based hypothesis
  - "force-align" - word start/end
- Foreach word start/end times
  - find "best" digit + silence
- Can use multi-class SVMs

- Simple approach to combining generative and discriminative models
  - related to acoustic code-breaking [26]
- Initial implementation uses a highly sub-optimal SVM combination scheme
  - use HMMs to find most confusable - simply apply SVMs in order
  - allowed a subset of confusions to be used

AURORA 2 Task

- AURORA 2 small vocabulary digit string recognition task
  - TIDIGITS databases used - utterances of one-seven digits
  - digits `zero-nine` plus `oh` used
  - clean training data 8440 utterances from 55 male and 55 female speakers
- Test Set A only considered for these experiments
  - four noise conditions N1-N4 (subway, babble, car and exhibition hall)
  - range of SNRS, only 00-20dB considered in this work
  - only 05-20dB used for SPR experiments
  - 1001 utterances used for evaluation in each test set
- Different MFCC parameterisation to standard AURORA MFCC coding
- Whole-word models, 16 emitting-states with 3 components per state.

VTS and SPR Performance

- Two VTS configurations used:
  - VTS$_0$ initial noise model for first and last 20 frames
  - VTS: noise-model estimated using hypotheses from VTS$_0$

SNR (dB)	System			
	—	SPR	VTS$_0$	VTS
20	5.30	1.80	2.62	1.66
15	16.27	2.81	3.75	2.30
10	40.35	5.40	7.03	4.37
05	69.75	12.89	14.75	11.04
00	87.30	—	32.90	29.75
Avg	43.79	—	12.21	9.82

- VTS works well - improved with noise estimation
  - VTS outperformed SPR - some level of speaker adaptation ...

SVM Rescoring

- SVMs trained on 9 out of the 16 noise conditions (N1/05dB not used)
  - only consider 05-20 dB (no 00dB SPR data)
  - 20 confusable digit pairs and all insertion/deletion confusions

SNR (dB)	System		
	—	SPR	+SVM
20	5.30	1.80	1.56
15	16.27	2.81	2.32
10	40.35	5.40	4.08
05	69.75	12.89	8.80
N1	—	5.44	3.54

- SVM generalises to unseen noise condition
  - N1 averaged over 05-20dB
  - largest gains from correctly handling large numbers of insertions

Noise Corrupted Resource Management

- Resource Management: artifical naval resource allocation task
  - ≈ 1000 word closed-vocabulary task
  - 109 training speakers, about 3.8 hours of training data
  - average performance over 3 test sets: Feb'89, Oct'89, Feb'91
  - cross-word state-clustered tri-phones, 6-components/state - see HTK recipe
- Data artificially corrupted by adding noise
  - operations rooms noise from NOISEX database added at 20dB (calculated using `NIST wavemd`)
- Task less suitable for combing with SVM rescoring

JUD and Correlation Modelling

Scheme	Σ_y	WER
—	—	38.2
VTS	diag	8.5
DPMC	diag	7.5
DPMC	full	6.9
VTS-JUD	diag	9.5
DPMC-JUD	full	7.9
PST	—	7.8

- VTS performance well on this task
  - DPMC out-performs VTS - note better dynamic parameter compensation
  - DPMC-full yields gains over diagonal case
- VTS and DPMC based JUD schemes shows degradations from full schemes
  - JUD far more efficient than VTS/DPMC
  - predictive semi-tied transforms (PST) work well

Conclusions

- Reviewed model-based compensation schemes
  - relies on ability to represent impact of noise on the clean speech
  - computationally expensive
  - works well on the artificial tasks described
- Discussed simple extensions to standard approaches
  - joint uncertainty decoding - handling computational cost
  - predictive linear transforms - handles changes in correlation
  - generative kernels - allows combination with discriminative models (SVMs)
- A number of extensions not discussed, or described in minimal detail
  - Algonquin and phase-sensitive models
  - adaptive training - allows schemes like CMN to be incorporated
  - performance on "real" data supplied by TREL - works well!

References

[1] SB Davis and P Mermelstein, "Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences," *IEEE Trans Acoustics Speech and Signal Processing*, vol. 28, no. 4, pp. 357–366, 1980.

[2] H Hermansky, "Perceptual Linear Predictive (PLP) Analysis of Speech," *J Acoustical Society of America*, vol. 87, no. 4, pp. 1738–1752, 1990.

[3] MJF Gales and SJ Young, "The application of hidden Markov models in speech recognition," *Foundations and Trends in Signal Processing*, vol. 1, no. 3, 2007.

[4] A Acero, *Acoustical and Environmental Robustness in Automatic Speech Recognition*, Kluwer Academic Publishers, 1993.

[5] TT Kristjannson, *Speech Recognition in Adverse Environments: a Probabilistic Approach*, Ph.D. thesis, Waterloo, Ontario, Canada, 2002.

[6] L Deng, J Droppo, and A Acero, "Enhancement of log mel power spectra of speech using a phase sensitive model the acoustic environemnt and sequential estimation of the corrupting noise," *Proc. IEEE Transactions on Speech and Audio Processing*, 2004.

[7] S Furui, "Speaker independent isolated word recognition using dynamic features of speech spectrum," *IEEE Transactions ASSP*, vol. 34, pp. 52–59, 1986.

[8] R. A. Gopinath, M. J. F. Gales, P. S. Gopalakrishnan, S. Balakrishnan-Aiyer, and M. A. Picheny, "Robust speech recognition in noise - performance of the IBM continuous speech recognizer on the ARPA noise spoke task," in *ARPA Workshop on Spoken Language System Technology*, 1995, pp. 127–130.

[9] X Huang, A Acero, and H-W Hon, *Spoken Language Processing*, Prentice Hall, 2001.

[10] J Droppo, L Deng, and A Acero, "Evaluation of the SPLICE Algorithm on the Aurora 2 Database," in *Proc. Eurospeech*, Aalberg, Denmark, 2001, pp. 217–220.

[11] L Neumeyer and M Weintraub, "Probabilistic Optimum Filtering for Robust Speech Recognition," in *Proc. ICASSP*, Adelaide, 1994.

[12] CJ Leggetter and PC Woodland, "Maximum Likelihood Linear Regression for Speaker Adaptation of Continuous Density Hidden Markov Models," *Computer Speech and Language*, vol. 9, no. 2, pp. 171–185, 1995.

[13] MJF Gales, "Maximum Likelihood Linear Transformations for HMM-Based Speech Recognition," *Computer Speech and Language*, vol. 12, pp. 75–98, 1998.

[14] MJF Gales, *Model-based Techniques for Noise Robust Speech Recognition*, Ph.D. thesis, Cambridge University, 1995.

[15] PJ Moreno, *Speech Recognition in Noisy Environments*, Ph.D. thesis, Carnegie Mellon University, 1996.

[16] A Acero, L Deng, T Kristjansson, and J Zhang, "HMM Adaptation using Vector Taylor Series for Noisy Speech Recognition," in *Proc ICSLP*, Beijing, China, 2000.

[17] H Liao, *Uncertainty Decoding For Noise Robust Speech Recognition*, Ph.D. thesis, Cambridge University, 2007.

[18] J Droppo, A Acero, and L Deng, "Uncertainty decoding with SPLICE for Noise Robust Speech Recognition," in *Proc ICASSP 02*, Orlando, Florida, 2002.

[19] H Liao and MJF Gales, "Issues with Uncertainty Decoding for Noise Robust Speech Recognition," in *Proc. ICSLP*, Pittsburgh, PA, 2006.

[20] MJF Gales, "Semi-tied Covariance Matrices For Hidden Markov Models," *IEEE Trans Speech and Audio Processing*, vol. 7, no. 3, pp. 272–281, 1999.

[21] MJF Gales and RC van Dalen, "Predictive linear transforms for noise robust speech recognition," in *Proceedings of the ASRU Workshop*, 2007, pp. 59–64.

[22] VN Vapnik, *Statistical Learning Theory*, John Wiley & Sons, 1998.

[23] N. Smith and M. Gales, "Speech recognition using SVMs," in *Advances in Neural Information Processing Systems 14*, T. G. Dietterich, S. Becker, and Z. Ghahramani, Eds. 2002, pp. 1197–1204, MIT Press.

[24] T Jaakkola and D Hausser, "Exploiting generative models in disciminative classifiers," in *Advances in Neural Information Processing Systems 11*, SA Solla and DA Cohn, Eds. 1999, pp. 487–493, MIT Press.

[25] MJF Gales and C Longworth, "Discriminative classifiers with generative kernels for noise robust ASR," Submitted to InterSPeech 2008.

[26] V Venkataramani, S Chakrabartty, and W Byrne, "Support vector machines for segmental minimum Bayes risk decoding of continuous speech," in *ASRU 2003*, 2003, pp. 13–18.