Robustness Techniques for Speech Recognition

Berlin Chen

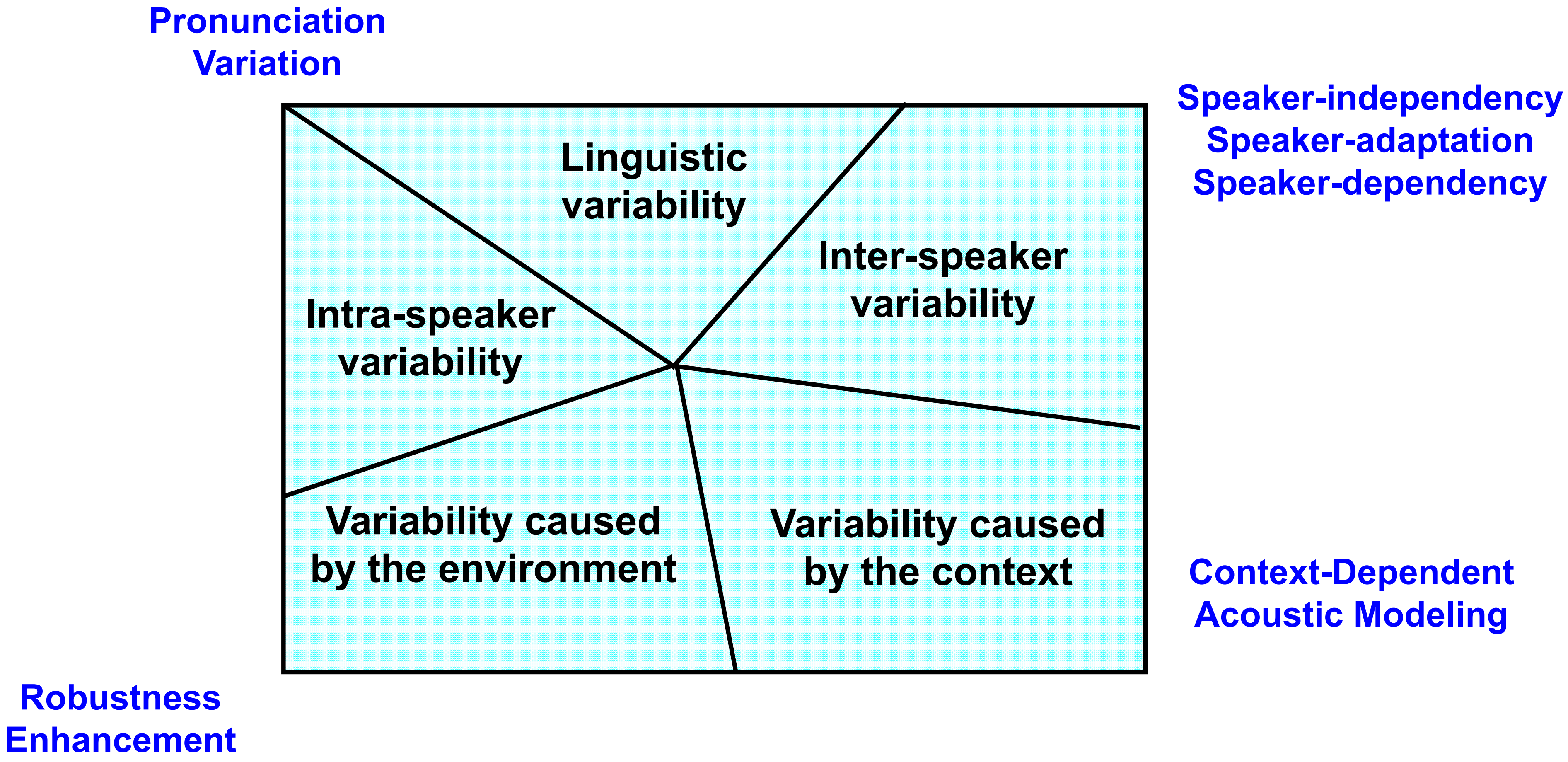
Department of Computer Science & Information Engineering
National Taiwan Normal University

References:

1. X. Huang et al. *Spoken Language Processing* (2001). Chapter 10
2. J. C. Junqua and J. P. Haton. *Robustness in Automatic Speech Recognition* (1996), Chapters 5, 8-9
3. T. F. Quatieri, *Discrete-Time Speech Signal Processing* (2002), Chapter 13
4. J. Droppo and A. Acero, "Environmental robustness," in Springer Handbook of Speech Processing, Springer, 2008, ch. 33, pp. 653–679.

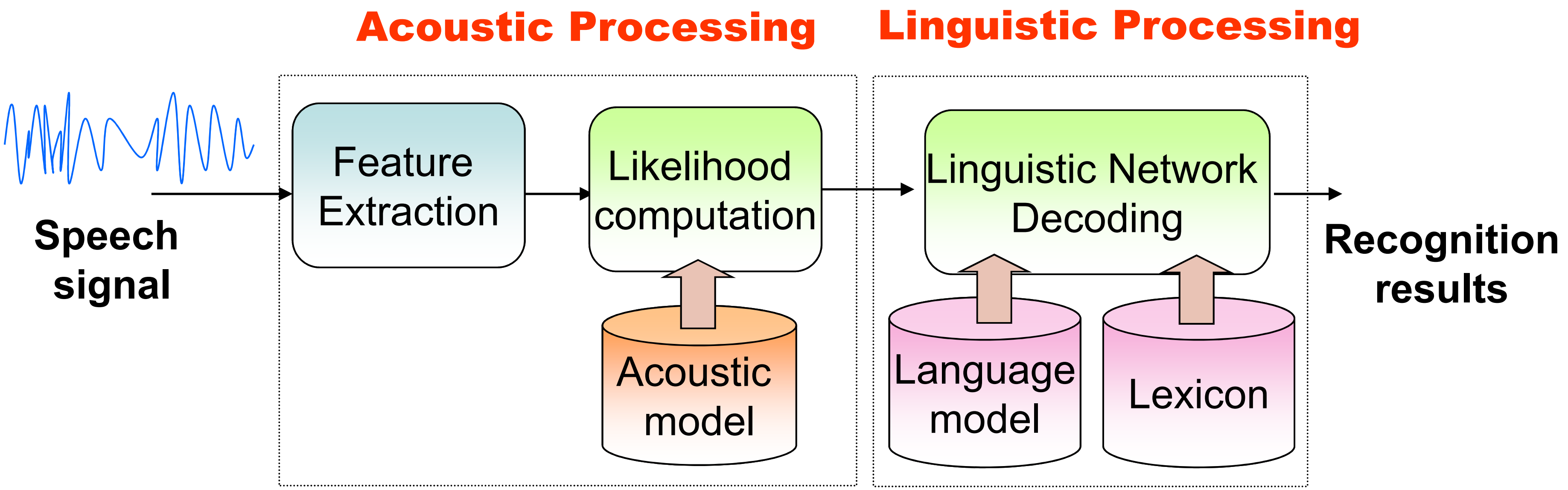
Introduction

- Classification of Speech Variability in Five Categories


Pronunciation Variation
Speaker-independency
Speaker-adaptation
Speaker-dependency
Linguistic variability
Intra-speaker variability
Inter-speaker variability
Variability caused by the environment
Variability caused by the context
Context-Dependent Acoustic Modeling
Robustness Enhancement

Introduction (cont.)

- The Diagram for Speech Recognition



- Importance of the *robustness* in speech recognition
  - Speech recognition systems have to operate in situations with uncontrollable acoustic environments
  - The recognition performance is often degraded due to the mismatch in the training and testing conditions
    - Varying environmental noises, different speaker characteristics (sex, age, dialects), different speaking modes (stylistic, Lombard effect), etc.

Introduction (cont.)

- If a speech recognition system's accuracy does not degrade very much under mismatch conditions, the system is called *robust*
  - ASR performance is rather uniform for SNRs greater than 25dB, but there is a very steep degradation as the noise level increases

$$25\,dB = 10\log_{10}\frac{E_s}{E_N} \Rightarrow \frac{E_s}{E_N} = 10^{2.5} \approx 316$$

    - Signal energy measured in time domain, e.g.:

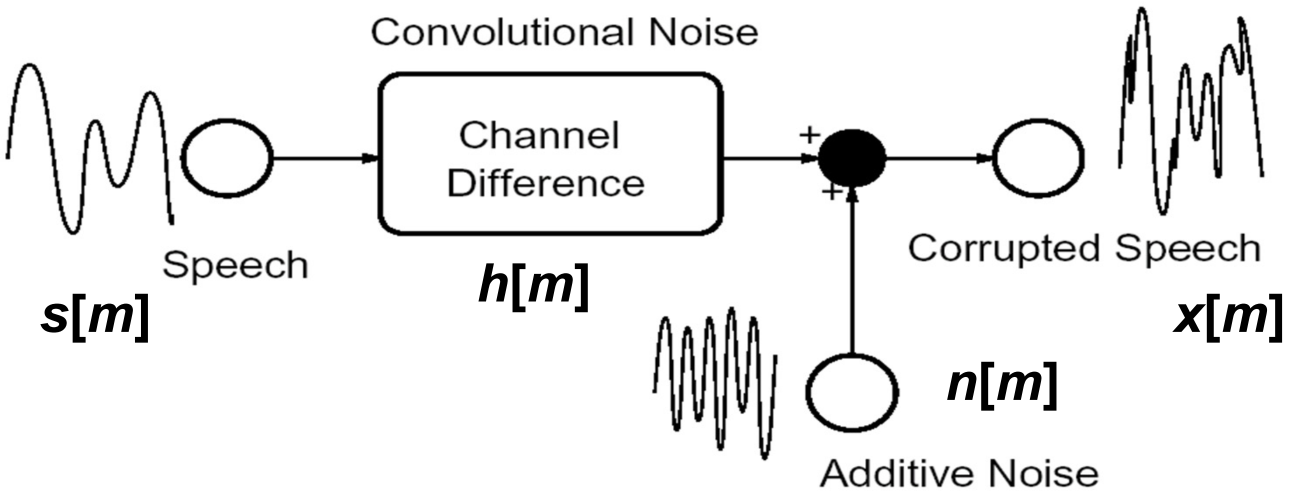
$$E_s = \frac{1}{T}\sum_{n=0}^{T-1} s[n]\times s^*[n]$$

- Various noises exist in varying real-world environments
  - Periodic, impulsive, or wide/narrow band

Introduction (cont.)

- Therefore, several possible robustness approaches have been developed to enhance the speech signal, its spectrum, and the acoustic models as well
  - Environmental compensation processing (feature-based)
  - Acoustic model adaptation (model-based)
  - Robust acoustic features (both model- and feature-based)
    - Or, inherently discriminative acoustic features

The Noise Types (1/2)



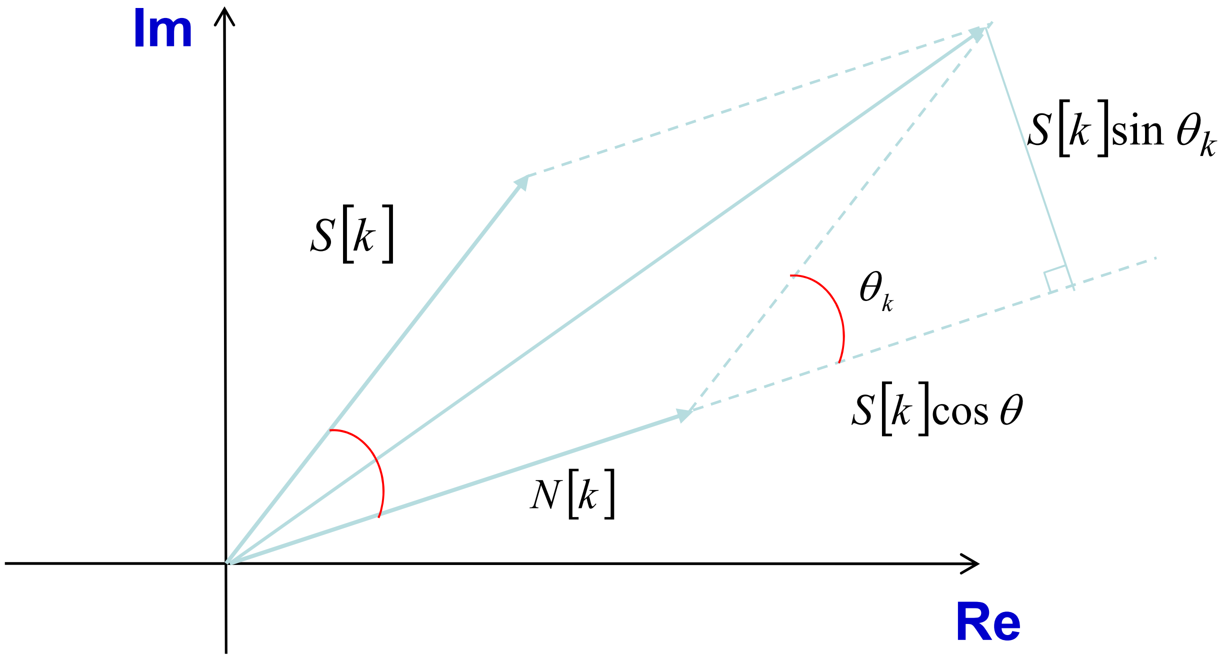
A model of the environment.

$$
\begin{aligned}
& x[m] = s[m] * h[m] + n[m] \\
& \Leftrightarrow\ X(\omega) = S(\omega) H(\omega) + N(\omega) \\
& \Leftrightarrow\ |X(\omega)|^2 = |S(\omega)|^2 |H(\omega)|^2 + |N(\omega)|^2 + 2\,\mathrm{Re}\{S(\omega) H(\omega) N^*(\omega)\} \\
& \qquad\qquad\ = |S(\omega)|^2 |H(\omega)|^2 + |N(\omega)|^2 + 2\underline{|S(\omega)||H(\omega)|}\underline{|N(\omega)|}\cos\theta_\omega \\
& \qquad\qquad\ \approx |S(\omega)|^2 |H(\omega)|^2 + |N(\omega)|^2
\end{aligned}
$$

or $P_X(\omega) = P_S(\omega) P_H(\omega) + P_N(\omega)$, $P(\cdot)$: power spectrum

or $S_{xx}(\omega) = S_{ss}(\omega) S_{hh}(\omega) + S_{nn}(\omega)$, $S_{--}(\cdot)$: spectrum

The Noise Types (2/2)



$$
\begin{aligned}
|X[k]|^2 &= |S[k] + N[k]|^2 \\
&= |S[k]|^2 \sin^2\theta_k + \left(|N[k]| + |S[k]|\cos\theta_k\right)^2 \\
&= |X[k]|^2 + |N[k]|^2 + 2|S[k]||N[k]|\cos\theta_k
\end{aligned}
$$

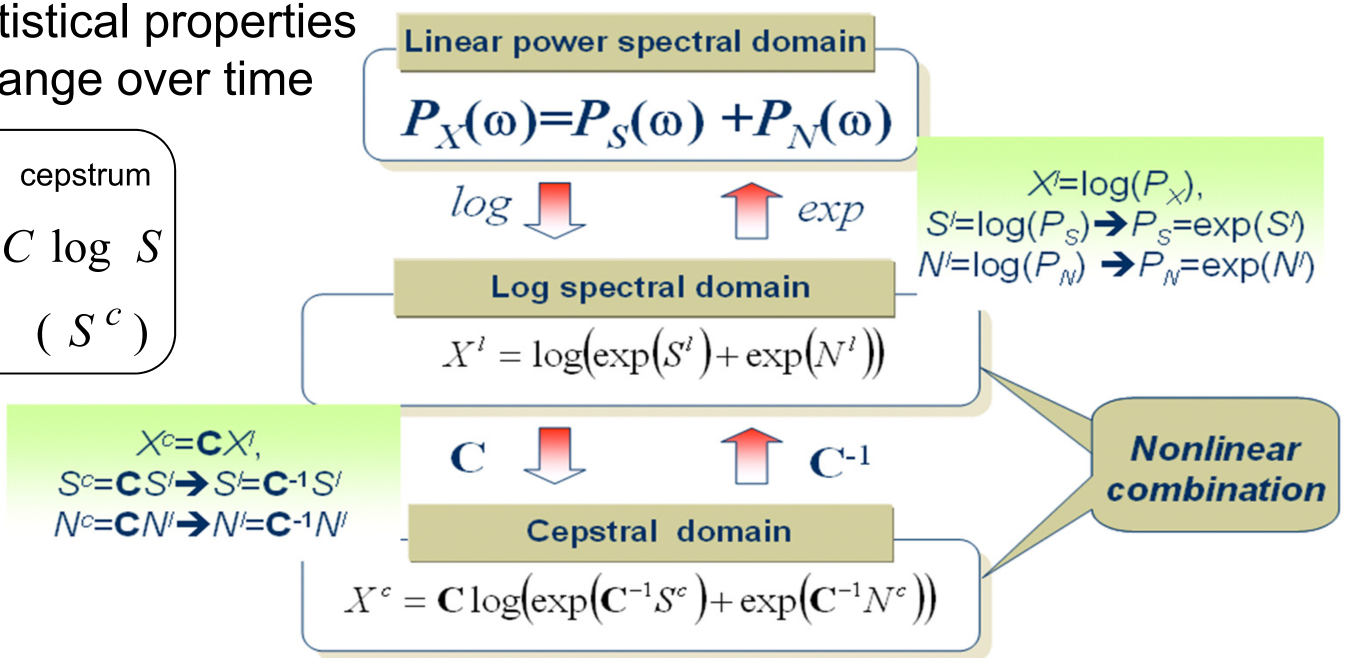
Additive Noises

- Additive noises can be stationary or non-stationary
  - Stationary noises
    - Such as computer fan, air conditioning, car noise: the power spectral density does not change over time (the above noises are also narrow-band noises)
  - Non-stationary noises
    - Machine gun, door slams, keyboard clicks, radio/TV, and other speakers' voices (babble noise, wide band nose, most difficult): the statistical properties change over time

power spectrum | log power spectrum | cepstrum

$$S \Rightarrow \log S \Rightarrow C \log S$$

$$(S^l) \qquad (S^c)$$



Linear power spectral domain

$$P_X(\omega)=P_S(\omega)+P_N(\omega)$$

$X^l=\log(P_X)$,
$S^l=\log(P_S) \rightarrow P_S=\exp(S^l)$
$N^l=\log(P_N) \rightarrow P_N=\exp(N^l)$

Log spectral domain

$$X^l=\log\left(\exp\left(S^l\right)+\exp\left(N^l\right)\right)$$

$X^c=\mathbf{C}X^l$,
$S^c=\mathbf{C}S^l \rightarrow S^l=\mathbf{C}^{-1}S^l$
$N^c=\mathbf{C}N^l \rightarrow N^l=\mathbf{C}^{-1}N^l$

Cepstral domain

$$X^c=\mathbf{C}\log\left(\exp\left(\mathbf{C}^{-1}S^c\right)+\exp\left(\mathbf{C}^{-1}N^c\right)\right)$$

Nonlinear combination

Additive Noises (cont.)

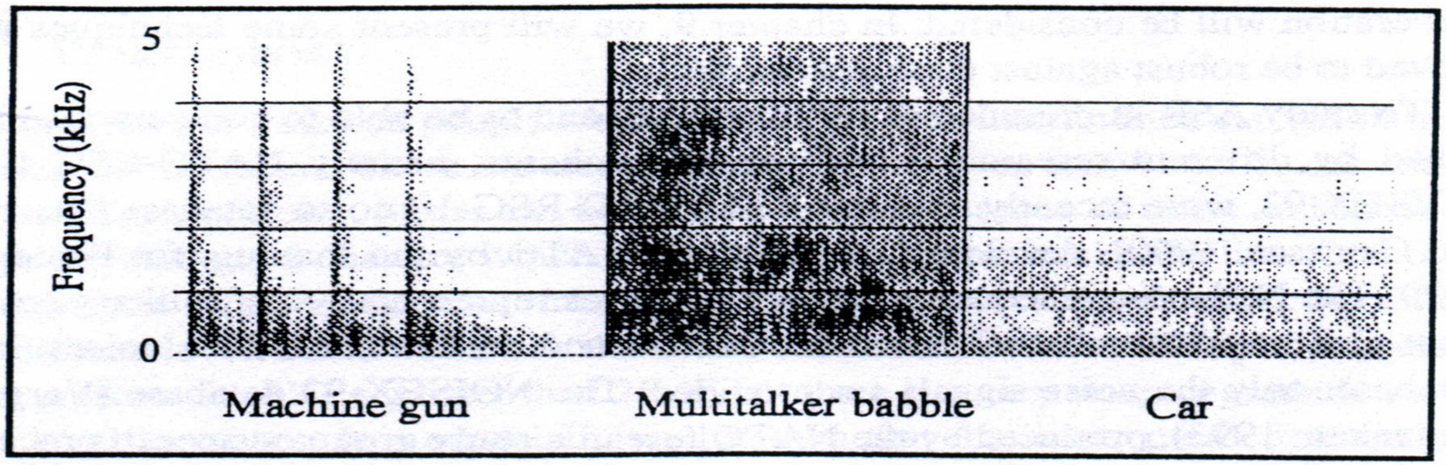


FIGURE 5.2 Spectrograms of three different types of noise.

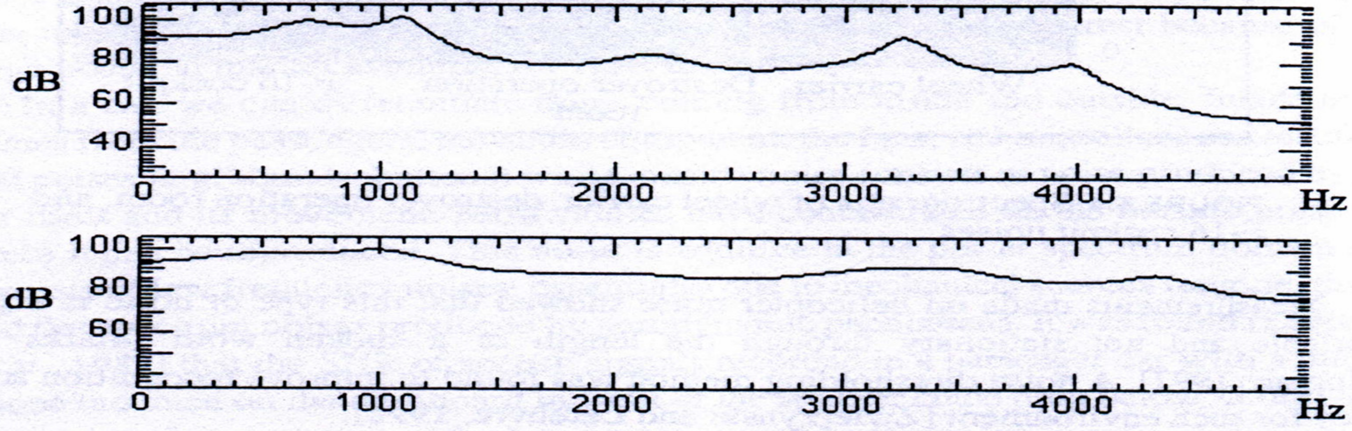


FIGURE 5.5 Effect of additive noise on the LPC log power spectrum of a frame in the vowel portion of the word "one" (in noise-free conditions (top), and with additive noise (bottom)).

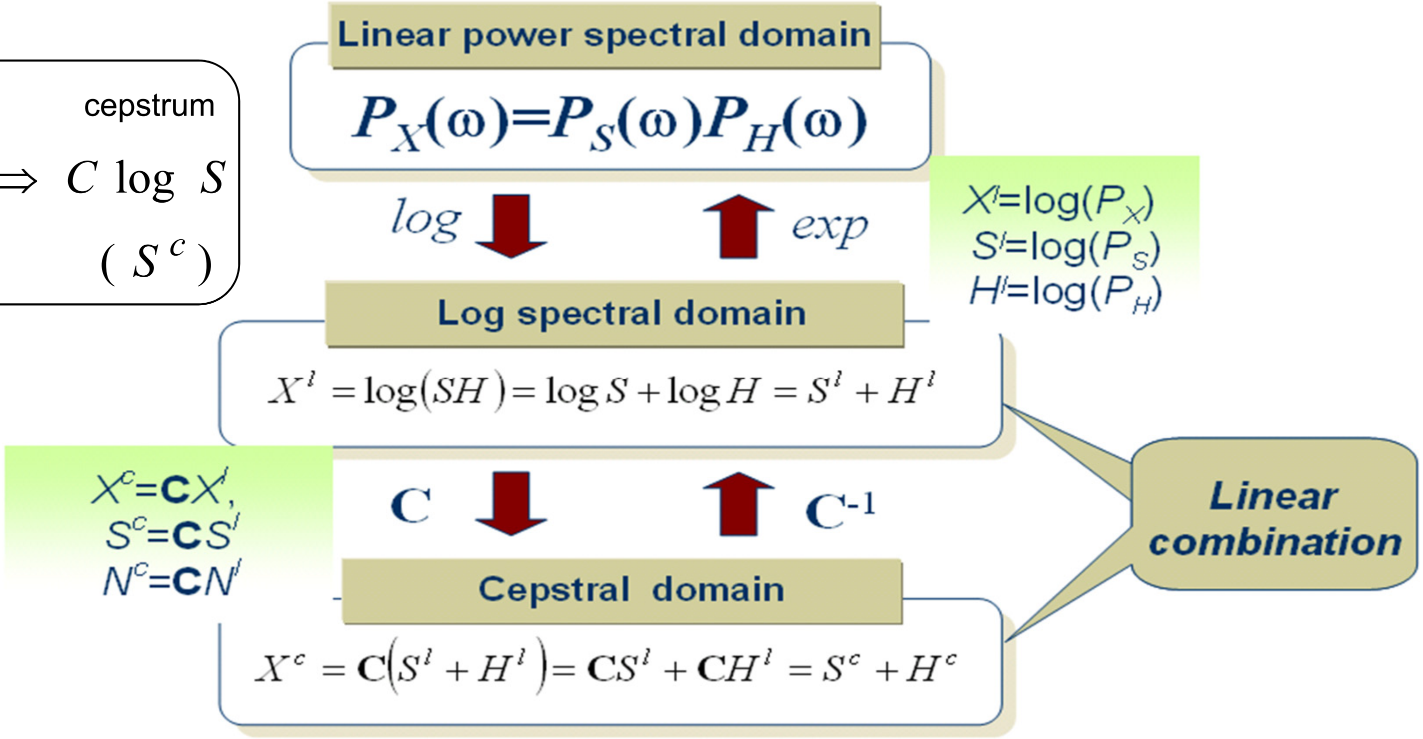
Convolutional Noises

- Convolutional noises are mainly resulted from channel distortion (sometimes called “channel noises”) and are stationary for most cases
  - Reverberation, the frequency response of microphone, transmission lines, etc.

power spectrum / log power spectrum / cepstrum

$$S \Rightarrow \log S \Rightarrow C \log S$$

$$(S^l) \qquad (S^c)$$



Linear power spectral domain

$$P_X(\omega)=P_S(\omega)P_H(\omega)$$

log / *exp*

$X^l=\log(P_X)$
$S^l=\log(P_S)$
$H^l=\log(P_H)$

Log spectral domain

$$X^l=\log(SH)=\log S+\log H=S^l+H^l$$

$\mathbf{C}$ / $\mathbf{C}^{-1}$

$X^c=\mathbf{C}X^l$,
$S^c=\mathbf{C}S^l$
$N^c=\mathbf{C}N^l$

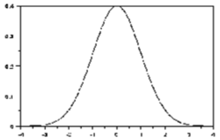
Cepstral domain

$$X^c=\mathbf{C}(S^l+H^l)=\mathbf{C}S^l+\mathbf{C}H^l=S^c+H^c$$

Linear combination

Noise Characteristics

- White Noise
  - The power spectrum is flat $S_{nn}(\omega) = q$,a condition equivalent to different samples being uncorrelated, $R_{nn}[m] = q\delta[m]$
  - White noise has a zero mean, but can have different distributions
  - We are often interested in the white Gaussian noise, as it resembles better the noise that tends to occur in practice



- Colored Noise
  - The spectrum is not flat (like the noise captured by a microphone)
  - *Pink noise*
    - A particular type of colored nose that has a low-pass nature, as it has more energy at the low frequencies and rolls off at high frequency
    - E.g., the noise generated by a computer fan, an air conditioner, or an automobile

Noise Characteristics (cont.)

- Musical Noise
  - Musical noise is short sinusoids (tones) randomly distributed over time and frequency
    - That occur due to, e.g., the drawback of original spectral subtraction technique and statistical inaccuracy in estimating noise magnitude spectrum
- Lombard effect
  - A phenomenon by which a speaker increases his vocal effect in the presence of background noise (the additive noise)
  - When a large amount of noise is present, the speaker tends to shout, which entails not only a high amplitude, but also often higher pitch, slightly different formants, and a different coloring (shape) of the spectrum
  - The vowel portion of the words will be overemphasized by the speakers

A Few Robustness Approaches

Three Basic Categories of Approaches

- Speech Enhancement Techniques
  - Eliminate or reduce the noisy effect on the speech signals, thus better accuracy with the originally trained models
    (Restore the clean speech signals or compensate for distortions)
  - *The feature part is modified while the model part remains unchanged*
- Model-based Noise Compensation Techniques
  - Adjust (changing) the recognition model parameters (*means and variances*) for better matching the noisy testing conditions
  - *The model part is modified while the feature part remains unchanged*
- Robust Parameters for Speech
  - Find robust representation of speech signals less influenced by additive or channel noise
  - *Both of the feature and model parts are changed*

Assumptions & Evaluations

- General Assumptions for the Noise
  - The noise is uncorrelated with the speech signal
  - The noise characteristics are fixed during the speech utterance or vary very slowly (the noise is said to be stationary)
    - The estimates of the noise characteristics can be obtained during non-speech activity
  - The noise is supposed to be additive or convolutional

- Performance Evaluations
  - Intelligibility, quality (**subjective** assessment)
  - Distortion between clean and recovered speech (**objective** assessment)
  - Speech recognition accuracy

Spectral Subtraction (SS)

S. F. Boll, 1979

- A Speech Enhancement Technique
- Estimate the magnitude (or the power) of clean speech by explicitly subtracting the noise magnitude (or the power) spectrum from the noisy magnitude (or power) spectrum
- Basic Assumption of Spectral Subtraction
  - The clean speech $s[m]$ is corrupted by additive noise $n[m]$
  - Different frequencies are uncorrelated from each other
  - $s[m]$ and $n[m]$ are statistically independent, so that the power spectrum of the noisy speech $x[m]$ can be expressed as:

    $$P_X(\omega)=P_S(\omega)+P_N(\omega)$$

  - To eliminate the additive noise: $P_S(\omega)=P_X(\omega)-P_N(\omega)$
  - We can obtain an estimate of $P_N(\omega)$ using the average period of *M* frames that are *known to be just noise*:

$$\hat{P}_N(\omega)=\frac{1}{M}\sum_{i=0}^{M-1}P_{N,i}(\omega)$$

frames

Spectral Subtraction (cont.)

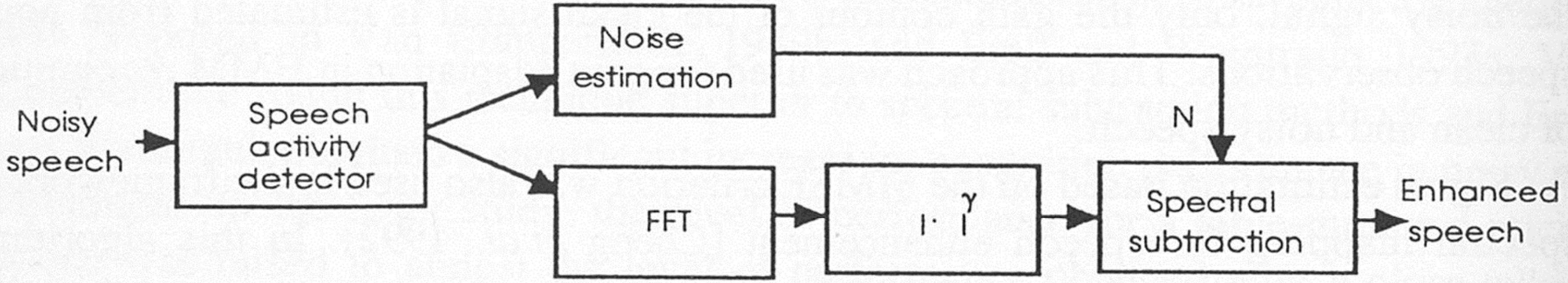


FIGURE 9.1 The spectral subtraction technique, $\gamma=1$ gives magnitude subtraction, $\gamma=2$ gives power subtraction.

- Problems of Spectral Subtraction
  - $s[m]$ and $n[m]$ are not statistically independent such that the cross term in power spectrum can not be eliminated
  - $\hat{P}_S(\omega)$ **is possibly less than zero**
  - Introduce “musical noise” when $P_X(\omega) \approx P_N(\omega)$
  - **Need a robust endpoint (speech/noise/silence) detector**

Spectral Subtraction (cont.)

- Modification: Nonlinear Spectral Subtraction (NSS)

$$\hat{P}_S(\omega) \equiv \begin{cases} \overline{P}_X(\omega) - \overline{P}_N(\omega), & \text{if } \overline{P}_X(\omega) \geq \overline{P}_N(\omega) \\ \overline{P}_N(\omega), & \text{otherwise} \end{cases}$$

$\overline{P}_X(\omega)$ and $\overline{P}_N(\omega)$: smoothed noisy and noise spectrum

or

$$\hat{P}_S(\omega) \equiv \begin{cases} \overline{P}_X(\omega) - \phi(\omega), & \text{if } \overline{P}_X(\omega) > \phi(\omega) + \beta \cdot \overline{P}_N(\omega) \\ \beta \cdot \overline{P}_N(\omega), & \text{otherwise} \end{cases}$$

$\overline{P}_X(\omega)$ and $\overline{P}_N(\omega)$: smoothed noisy and noise spectrum

$\phi(\omega)$: a non - linear function according to SNR

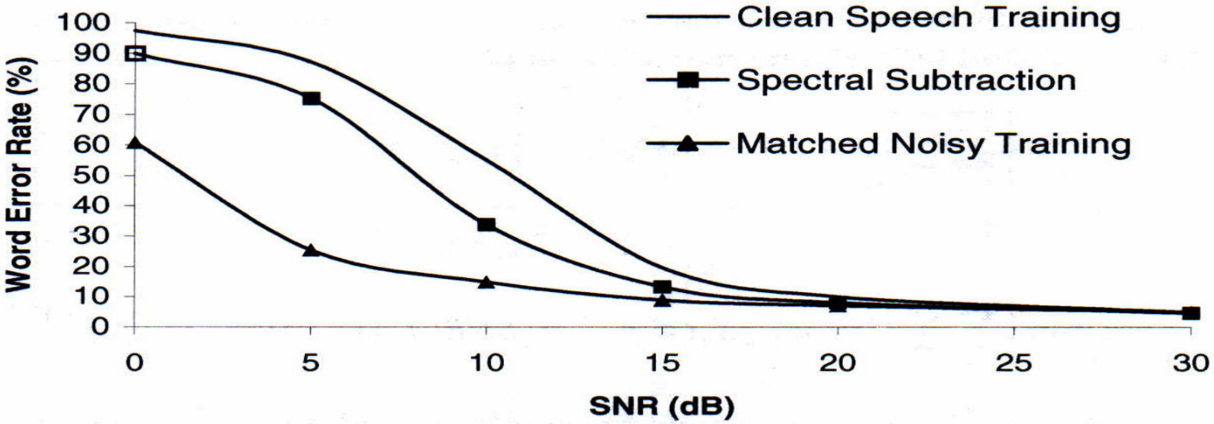


Figure 10.28 Word error rate as a function of SNR (dB) using Whisper on the *Wall Street Journal* 5000-word dictation task. White noise was added at different SNRs. The solid line represents the baseline system trained with clean speech, the line with squares the use of spectral subtraction with the previous clean HMMs. They are compared to a system trained on the same speech with the same SNR as the speech tested on.

Spectral Subtraction (cont.)

- Spectral Subtraction can be viewed as a filtering operation

Power Spectrum

$$
\begin{aligned}
\hat{P}_S(\omega) &= P_X(\omega) - P_N(\omega) \\
&= P_X(\omega)\left[1 - \frac{P_N(\omega)}{P_X(\omega)}\right] = P_X(\omega)\left[\frac{P_S(\omega)}{P_S(\omega) + P_N(\omega)}\right] \quad (\text{supposed that } P_X(\omega) \approx P_S(\omega) + P_N(\omega)) \\
&= P_X(\omega)\left[1 + \frac{1}{R(\omega)}\right]^{-1} \qquad \left(R(\omega) = \frac{P_S(\omega)}{P_N(\omega)} : \text{instantane ous SNR } \right)
\end{aligned}
$$

$\therefore$ The time varying suppressio n filter is given approximat ely by :

$$
H(\omega) = \left[1 + \frac{1}{R(\omega)}\right]^{-1/2}
$$

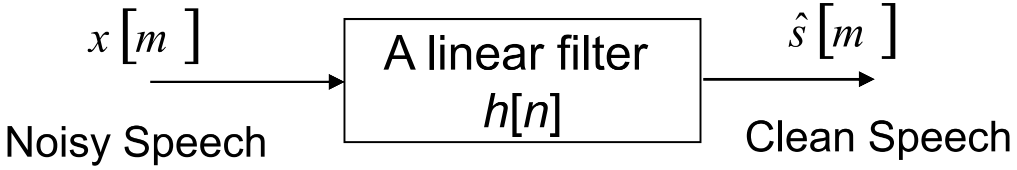
Spectrum Domain

Wiener Filtering

- A Speech Enhancement Technique
- From the Statistical Point of View
  - The process $\boldsymbol{x}[m]$ is the sum of the random process $\boldsymbol{s}[m]$ and the additive noise process $\boldsymbol{n}[m]$

    $$\boldsymbol{x}[m] = \boldsymbol{s}[m] + \boldsymbol{n}[m]$$

  - Find a linear estimate $\hat{\boldsymbol{s}}[m]$ in terms of the process $\boldsymbol{x}[m]$:
    - Or to find a linear filter $h[m]$ such that the sequence $\hat{s}[m] = x[m] * h[m]$ minimizes the expected value of $(\hat{s}[m] - s[m])^2$



$x[m]$

A linear filter $h[n]$

$\hat{s}[m]$

Noisy Speech

Clean Speech

$$\begin{aligned} \hat{s}[m] &= x[m] * h[m] \\ &= \sum_{l=-\infty}^{\infty} h[l]\, x[m-l] \end{aligned}$$

Wiener Filtering (cont.)

- Minimize the expectation of the squared error (MMSE estimate)

$$Minimize\ F = E\left\{\left[s[m] - \sum_{l=-\infty}^{\infty} h[l]x[m-l]\right]^2\right\}$$

$$\forall_k \frac{\partial F}{h[k]} = 0$$

$$\Rightarrow \forall_k\ s[m]x[m-k] = \left(\sum_{l=-\infty}^{\infty} h[l]x[m-l]\right)x[m-k]$$

Take summation for ***m***

$$\Rightarrow \sum_{m=-\infty}^{\infty} s[m]x[m-k] = \sum_{l=-\infty}^{\infty} h[l] \sum_{m=-\infty}^{\infty} x[m-l]x[m-k]$$

$$\Rightarrow \sum_{m=-\infty}^{\infty} s[m](s[m-k] + n[m-k]) = \sum_{l=-\infty}^{\infty} h[l] \sum_{m=-\infty}^{\infty} x[m-l]x[m-k]$$

$s[m]$ and $n[m]$ are statistically independent!

$$\Rightarrow \sum_{m=-\infty}^{\infty} s[m]s[m-k] + \cancel{\sum_{m=-\infty}^{\infty} s[m]n[m-k]} = \sum_{l=-\infty}^{\infty} h[l]R_x[k-l]$$

$$\Rightarrow R_s[k] = h[k] * R_x[k]$$

$$\Rightarrow S_{ss}(\omega) = H(\omega)S_{xx}(\omega)$$

$R_s[n]$ and $R_x[n]$: are respective ly the autocorrel ation sequences of $s[n]$ and $x[n]$

Take Fourier transformm

Wiener Filtering (cont.)

- Minimize the expectation of the squared error (MMSE estimate)

$$\because S_{ss}(\omega) = H(\omega) S_{xx}(\omega)$$

$$\Rightarrow H(\omega) = \frac{S_{ss}(\omega)}{S_{xx}(\omega)} = \frac{S_{ss}(\omega)}{S_{ss}(\omega) + S_{nn}(\omega)}$$, is called the noncausal Wiener filter

(where $S_{xx}(\omega) = S_{ss}(\omega) + S_{nn}(\omega)$)

Wiener Filtering (cont.)

- The time varying Wiener Filter also can be expressed in a similar form as the spectral subtraction

$$H(\omega) = \frac{S_{ss}(\omega)}{S_{ss}(\omega) + S_{nn}(\omega)} = \frac{P_S(\omega)}{P_S(\omega) + P_N(\omega)}$$

$$= \left[1 + \frac{P_N(\omega)}{P_S(\omega)}\right]^{-1} = \left[1 + \frac{1}{R(\omega)}\right]^{-1}, \quad \left(R(\omega) = \frac{P_S(\omega)}{P_N(\omega)} : \text{instantane ous } SNR \right)$$

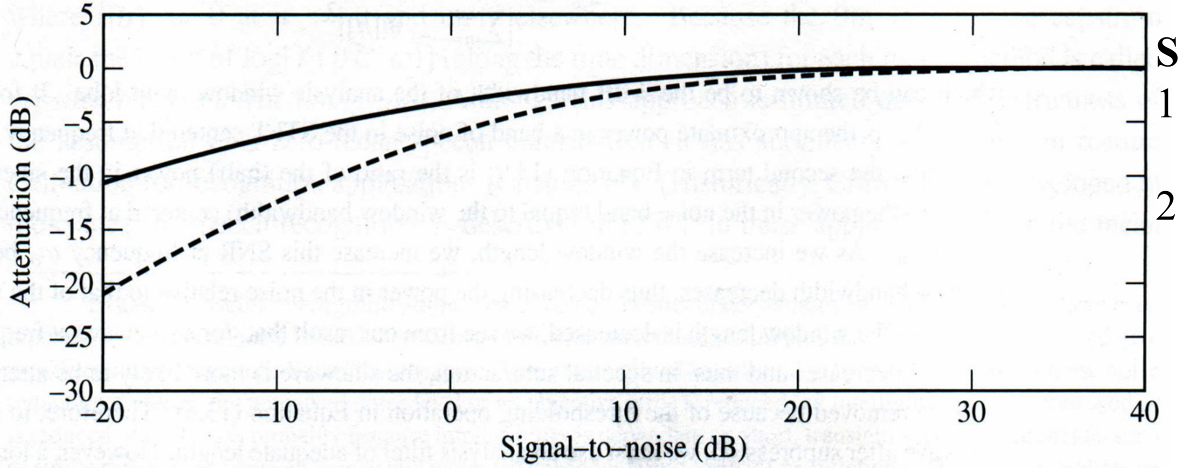


Figure 13.1 Comparison of suppression curves for spectral subtraction (solid line) and the Wiener filter (dashed line) as a function of the instantaneous SNR.

$$10 \log \frac{P_S(\omega)}{P_N(\omega)}$$

SS vs. Wiener Filter:

1. Wiener filter has stronger attenuation at low SNR region
2. Wiener filter does not invoke an absolute thresholding

Wiener Filtering (cont.)

- Wiener Filtering can be realized only if we know the power spectra of both the noise and the signal
  - A chicken-and-egg problem

- Approach - I : Ephraim(1992) proposed the use of an HMM where, if we know the current frame falls under, we can use its mean spectrum as $S_{ss}(\omega)$ or $P_S(\omega)$
  - In practice, we do not know what state each frame falls into either
    - Weight the filters for each state by a posterior probability that frame falls into each state

Wiener Filtering (cont.)

- Approach - II :
  - The background/noise is stationary and its power spectrum can be estimated by averaging spectra over a known background region
  - For the non-stationary speech signal, its time-varying power spectrum can be estimated using the past Wiener filter (of previous frame)

$$\hat{P}_S(t,\omega) = P_X(t,\omega)H(t-1,\omega), \ (t : \text{frame index}, H(\cdot) : \text{Wiener filter})$$

$$\therefore H(t,\omega) = \frac{\hat{P}_S(t,\omega)}{\hat{P}_S(t,\omega) + P_N(\omega)}$$

$$\widetilde{P}_S(t,\omega) = P_X(t,\omega)H(t,\omega)$$

    - The initial estimate of the speech spectrum can be derived from spectral subtraction
  - Sometimes introduce musical noise

Wiener Filtering (cont.)

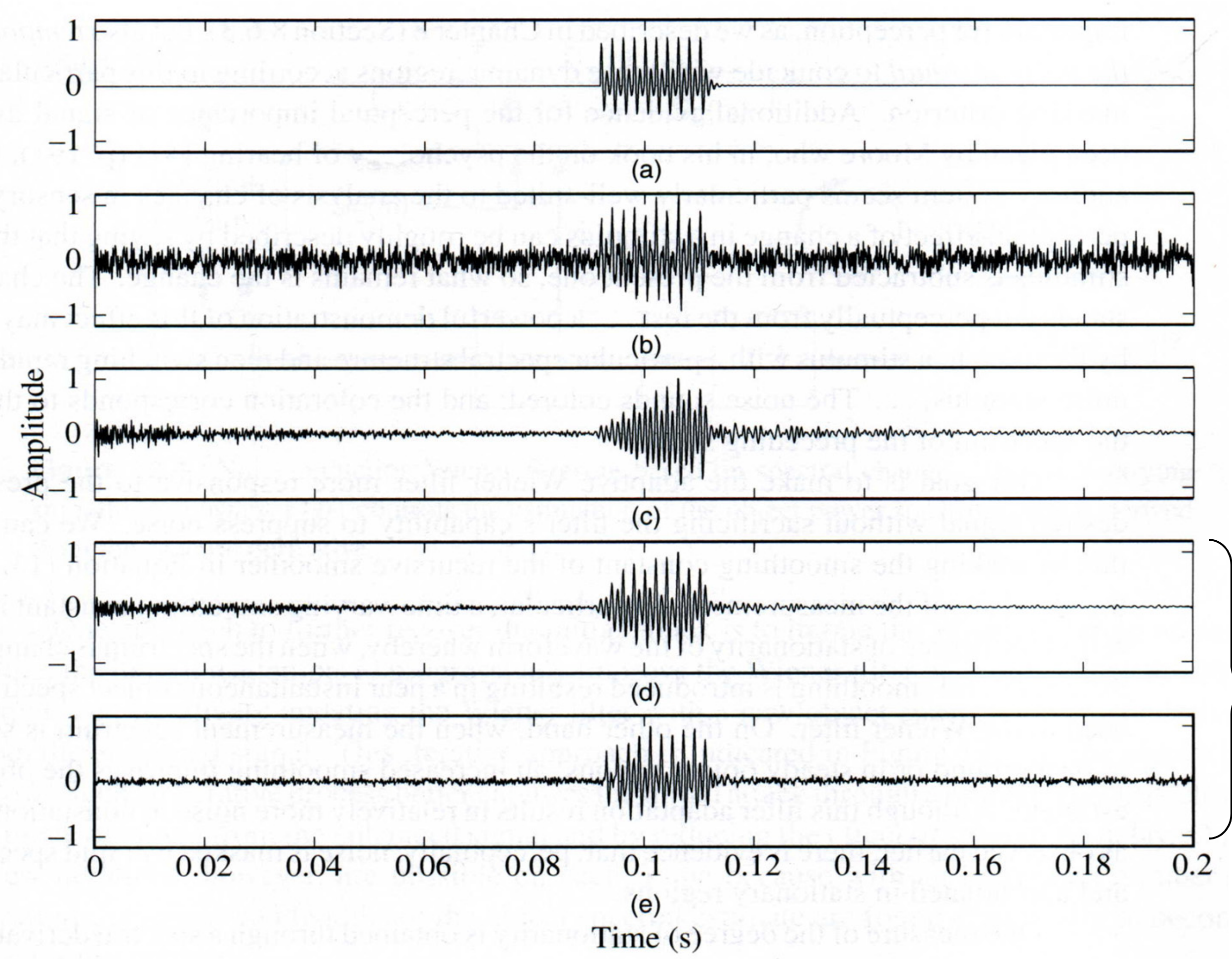
- Approach - III :
  - Slow down the rapid frame-to-frame movement of the object speech power spectrum estimate by apply temporal smoothing

$$\overset{\frown}{P}_S(t,\omega) = \alpha \cdot \widetilde{P}_S(t-1,\omega) + (1-\alpha) \cdot \hat{P}_S(t,\omega)$$

Then use $\overset{\frown}{P}_S(t,\omega)$ to replace $\hat{P}_S(t,\omega)$ in

$$H(t,\omega) = \frac{\hat{P}_S(t,\omega)}{\hat{P}_S(t,\omega) + P_N(\omega)} \quad \Rightarrow \quad H(t,\omega) = \frac{\overset{\frown}{P}_{\mathrm{S}}(t,\omega)}{\overset{\frown}{P}_{\mathrm{S}}(t,\omega) + P_N(\omega)}$$

Wiener Filtering (cont.)



Clean Speech

Noisy Speech

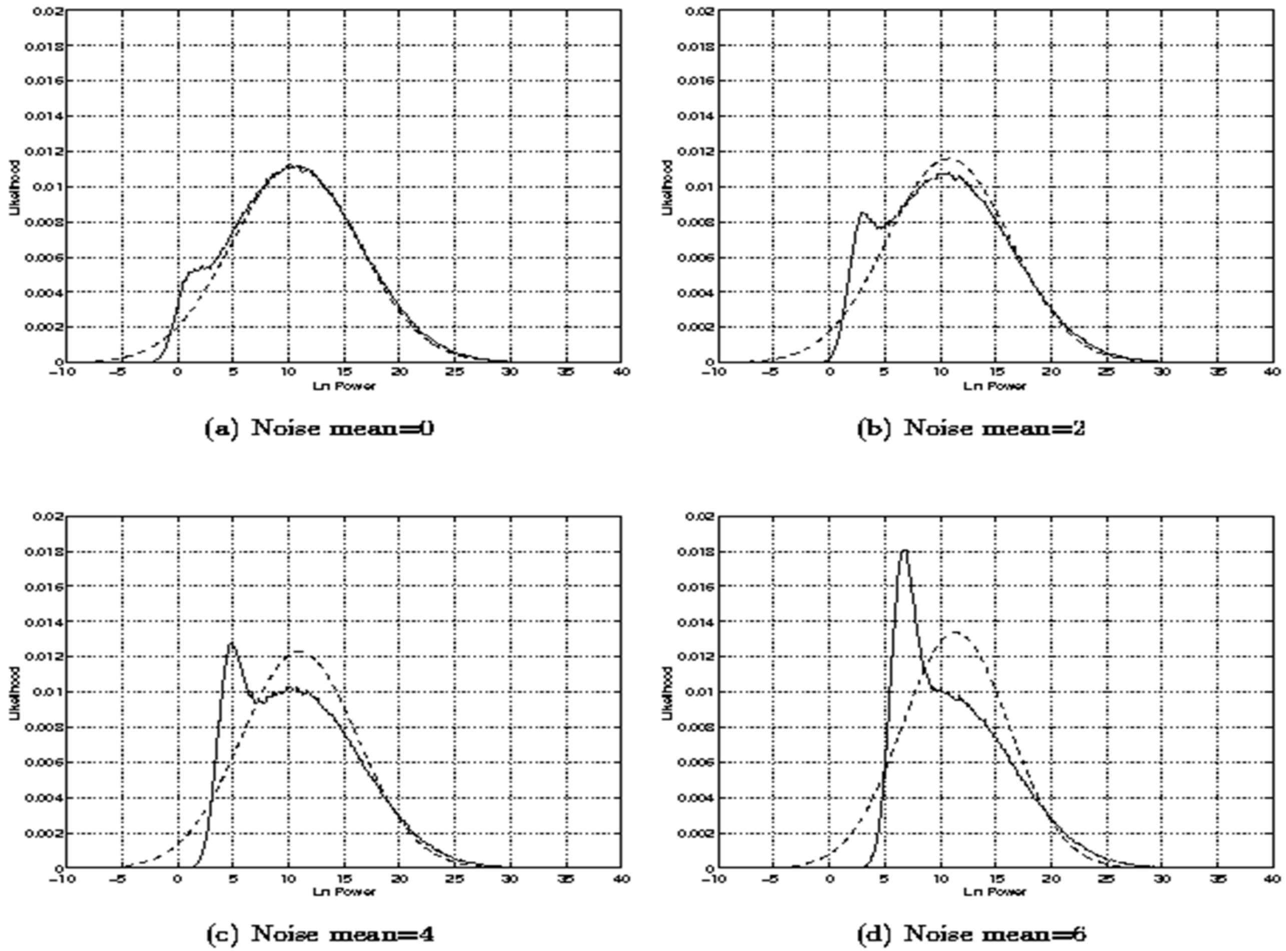
Enhanced Noise Speech Using Approach – III

$\alpha = 0.85$

Other more complicate Wiener filters

Figure 13.3 Enhancement by adaptive Wiener filtering of a train of closely-spaced decaying sinewaves in 10 dB of additive white Gaussian noise: (a) original clean object signal; (b) original noisy signal; (c) enhanced signal without use of spectral change; (d) enhanced signal with use of spectral change; (e) enhanced signal using spectral change, the iterative filter estimate (2 iterations), and background adaptation.

The Effectives of Active Noise



(a) Noise mean=0

(b) Noise mean=2

(c) Noise mean=4

(d) Noise mean=6

Figure 4.1: Plots of "corrupted-speech" distribution (solid), and maximum likelihood Gaussian distribution (dashed)

Cepstral Mean Normalization (CMN)

- A Speech Enhancement Technique and sometimes called *Cepstral Mean Subtraction* (CMS)
- CMN is a powerful and simple technique designed to handle convolutional (*Time-invariant linear filtering*) *distortions*

$$x[n] = s[n] * h[n]$$ Time Domain

$$X(\omega) = S(\omega)H(\omega)$$ Spectral Domain

$$X^l = \log|SH|^2 = \log|S|^2 + \log|H|^2 = S^l + H^l$$ Log Power Spectral Domain

$$CX^l = C(S^l + H^l) = CS^l + CH^l$$ Cepstral Domain

$$\overline{CS^l} = \frac{1}{T}\sum_{t=0}^{T-1} CS^l{}_t \text{ and } \overline{CX^l} = \frac{1}{T}\sum_{t=0}^{T-1}\left(CS^l{}_t + CH^l\right) = \overline{CS^l} + CH^l$$

if the training and testing speech materials were recored from two different channels

$$\text{Training}: CX(1)^l = C\left(S^l + H(1)^l\right) = CS^l + CH(1)^l, \text{Testing}: CX(2)^l = C\left(S^l + H(2)^l\right) = CS^l + CH(2)^l$$

$$CX(1)^l - \overline{CX(1)^l} = CS^l - \overline{CS^l}$$

$$CX(2)^l - \overline{CX(2)^l} = CS^l - \overline{CS^l}$$

The spectral characteristics of the microphone and room acoustics thus can be removed !

Can be eliminated if the assumption of zero-mean speech contribution!

Cepstral Mean Normalization (cont.)

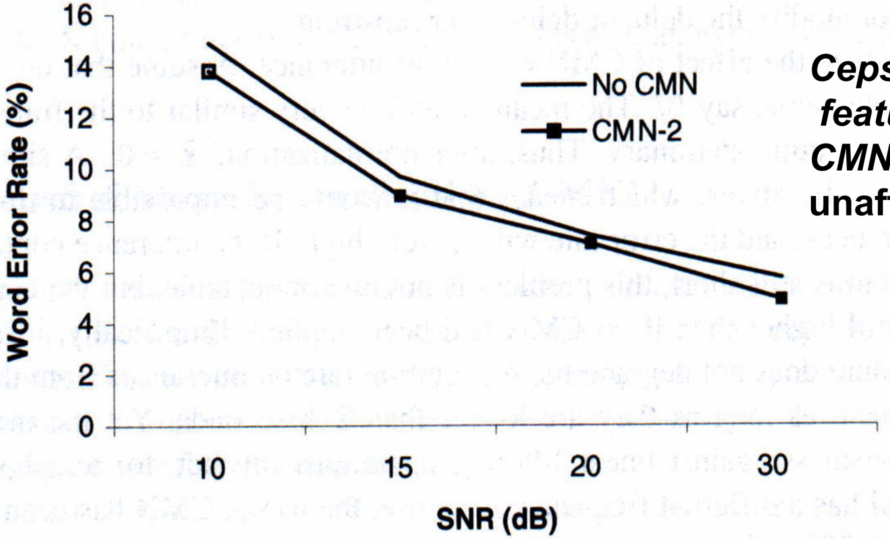
- Some Findings
  - Interesting, CMN has been found effective even the testing and training utterances are within the same microphone and environment
    - Variations for the distance between the mouth and the microphone for different utterances and speakers
  - Be careful that the **duration/period** used to estimate the mean of noisy speech
    - Why?
      - Problematic when the acoustic feature vectors are almost identical within the selected time period

Cepstral Mean Normalization (cont.)

- Performance
  - For telephone recordings, where each call has different frequency response, the use of CMN has been shown to provide as much as 30 % relative decrease in error rate
  - When a system is trained on one microphone and tested on another, CMN can provide significant robustness

Cepstral Mean Normalization (cont.)

- CMN has been shown to improve the robustness not only to varying channels but also to the noise
  - White noise added at different SNRs
  - System trained with speech with the same SNR (matched Condition)

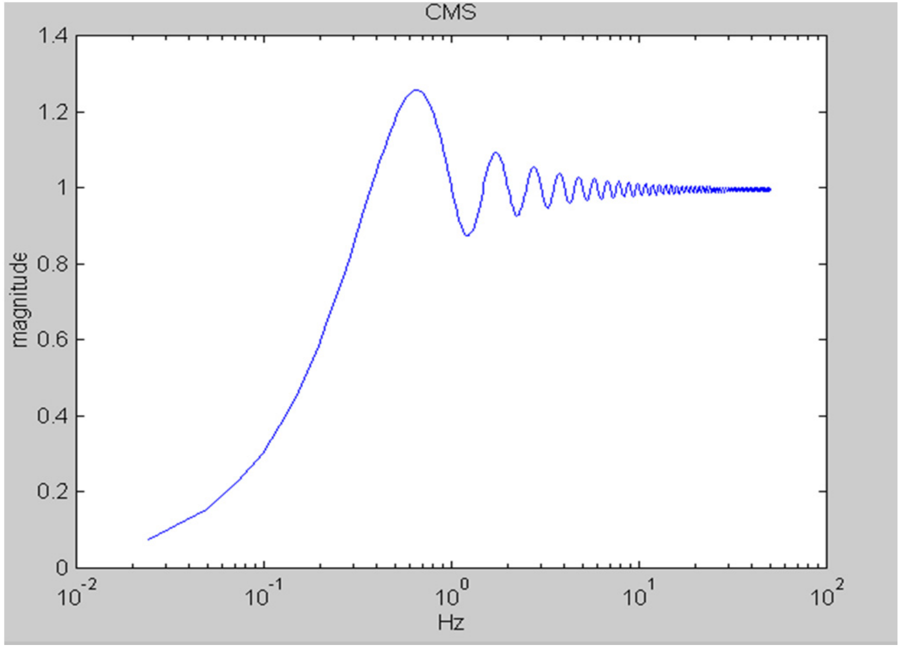


***Cepstral delta and delta-delta features are computed prior to the CMN operation* so that they are unaffected.**

Figure 10.30 Word error rate as a function of SNR (dB) for both no CMN and CMN-2 [5]. White noise was added at different SNRs and the system was trained with speech with the same SNR. The Whisper system is used on the 5000-word *Wall Street Journal* task using a bigram language model.

Cepstral Mean Normalization (cont.)

- From the other perspective
  - We can interpret CMN as the operation of subtracting a low-pass temporal filter $d[n]$, where all the T coefficients are identical and equal to $1/T$, which is a high-pass temporal filter
  - Alleviate the effect of conventional noise introduced in the channel



Temporal (Modulation) Frequency

Cepstral Mean Normalization (cont.)

- Real-time Cepstral Normalization
  - CMN requires the complete utterance to compute the cepstral mean; thus, it cannot be used in a real-time system, and an approximation needs to be used
  - Based on the above perspective, we can implement other types of high-pass filters

$$\overline{CX^{l}{}_{t}} = \alpha \cdot CX^{l}{}_{t} + (1-\alpha) \cdot \overline{CX^{l}{}_{t-1}} \text{ , } (\overline{CX^{l}{}_{t}} : \text{cepstral mean})$$

Histogram EQualization (HEQ)

- HEQ has its roots in the assumption that the transformed speech feature distributions of the test (or noisy) data should be identical to that of the training (or reference) data y
  - Find a transformation function $F(x)$ converts x to y satisfying

$$p_{Train}(y) = p_{Test}(x)\frac{dx}{dy} = p_{Test}\left(F^{-1}(y)\right)\frac{dF^{-1}(y)}{dy}$$

$$\begin{aligned} & y = F(x) \\ & \Rightarrow x = F^{-1}(y) \end{aligned}$$

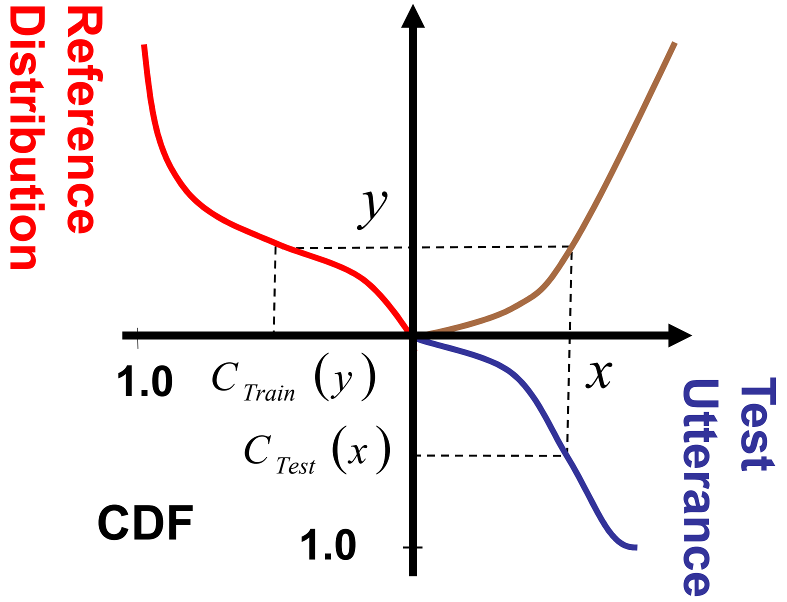
  - Therefore, the relationship between the cumulative probability density functions (CDFs) respectively associated with the test and training speech is

$$\begin{aligned} C_{test}(x) &= \int_{-\infty}^{x} p_{Test}(x')\,dx' \\ &= \int_{-\infty}^{F(x)} p_{Test}(F^{-1}(y'))\frac{dF^{-1}(y')}{dy'}\,dy' \\ &= \int_{-\infty}^{y} p_{Train}(y')\,dy'\Big|_{y=F(x)} \\ &= C_{Train}(y) \end{aligned}$$

Clean	Noisy
0.50	0.53
2.10	1.95
1.20	1.40
-3.50	-3.20
4.31	4.47

Histogram Equalization (Cont.)

- Convert the distribution of each feature vector component of the test speech into a predefined target distribution which corresponds to that of the training speech
  - Attempt not only to match speech feature means/variances, but also to completely match the feature distribution of the training and test data



RASTA Temporal Filter

Hyneck Hermansky, 1991

- **A Speech Enhancement Technique**
- **RASTA *(Relative Spectral)***

Assumption

- The linguistic message is coded into movements of the vocal tract (i.e., the change of spectral characteristics)
- The rate of change of non-linguistic components in speech often lies outside the typical rate of change of the vocal tact shape
  - E.g. fix or slow time-varying linear communication channels
- A great sensitivity of human hearing to modulation frequencies around 4Hz than to lower or higher modulation frequencies

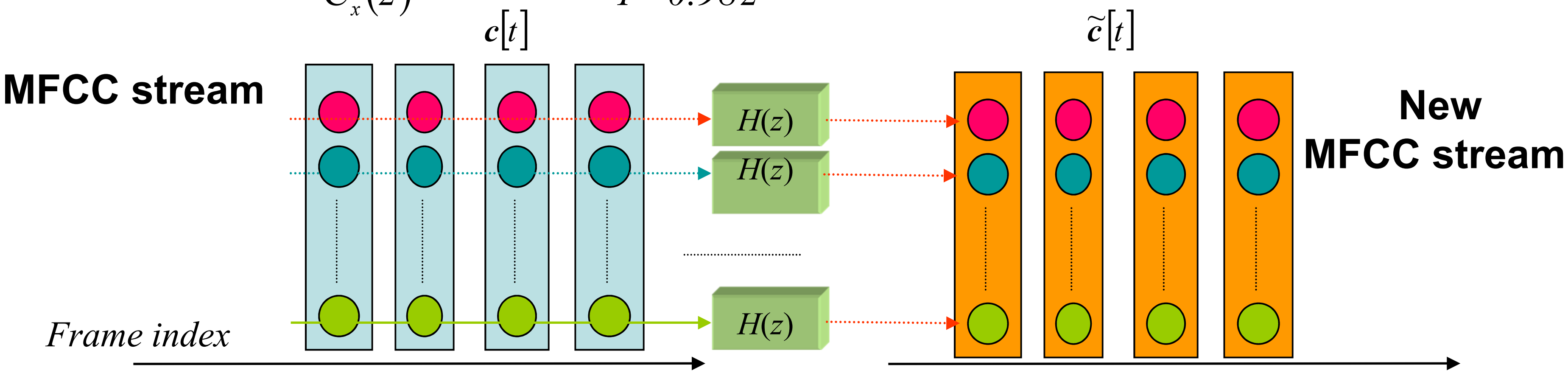
Effect

- RASTA Suppresses the spectral components that change more *slowly* or *quickly* than the typical rate of change of speech

RASTA Temporal Filter (cont.)

- The IIR transfer function

$$H(z)=\frac{\tilde{C}_x(z)}{C_x(z)}=0.1z^{4}\cdot\frac{2+z^{-1}-z^{-3}-2z^{-4}}{1-0.98z^{-1}}$$



$\boldsymbol{c}[t]$

$\tilde{\boldsymbol{c}}[t]$

MFCC stream

$H(z)$

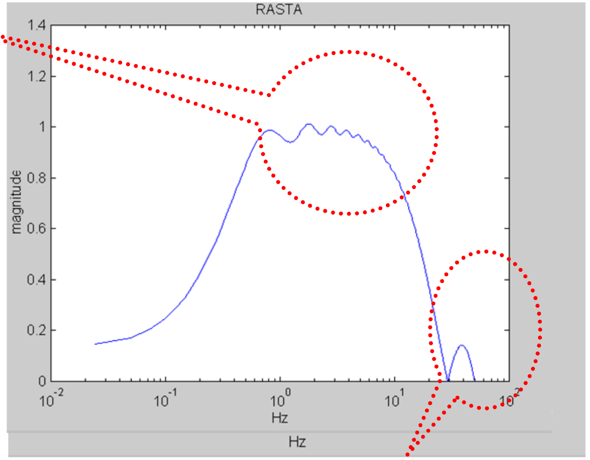
$H(z)$

$H(z)$

New MFCC stream

Frame index

RASTA has a peak at about 4Hz (modulation frequency)



modulation frequency 100 Hz

- An other version

$$H(z)=0.1\cdot\frac{2+z^{-1}-z^{-3}-2z^{-4}}{1-0.98z^{-1}}$$

$$\begin{aligned}\tilde{\boldsymbol{c}}[t]=0.98\cdot\tilde{\boldsymbol{c}}[t-1]+0.2\cdot\boldsymbol{c}[t]+0.1\cdot\boldsymbol{c}[t-1]\\-0.1\cdot\boldsymbol{c}[t-3]+0.2\cdot\boldsymbol{c}[t-4]\end{aligned}$$

Retraining on Corrupted Speech

- A Model-based Noise Compensation Technique
- Matched-Conditions Training
  - Take a noise waveform from the new environment, add it to all the utterance in the training database, and retrain the system
  - If the noise characteristics are known ahead of time, this method allow as to adapt the model to the new environment with relatively small amount of data from the new environment, yet use a large amount of training data

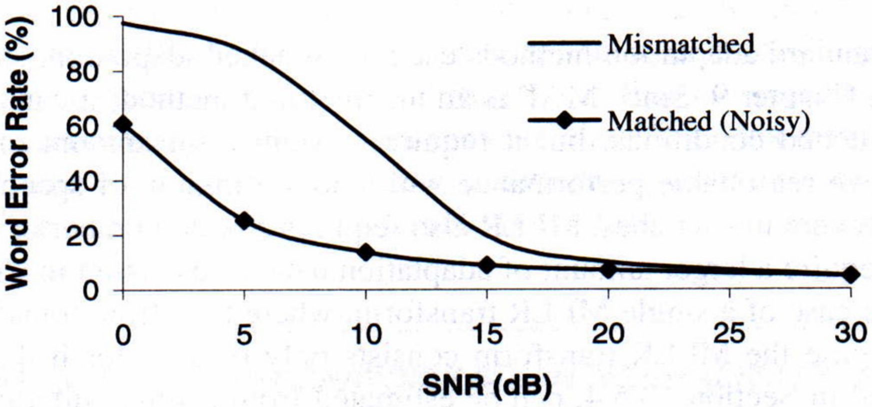


Figure 10.31 Word error rate as a function of the testing data SNR (dB) for Whisper trained on clean data and a system trained on noisy data at the same SNR as the testing set as in Figure 10.30. White noise at different SNRs is added.

Retraining on Corrupted Speech (cont.)

- Multi-style Training
  - Create a number of artificial acoustical environments by corrupting the clean training database with noise samples of varying levels (30dB, 20dB, etc.) and types (white, babble, etc.), as well as varying the channels
  - All those waveforms (copies of training database) from multiple acoustical environments can be used in training

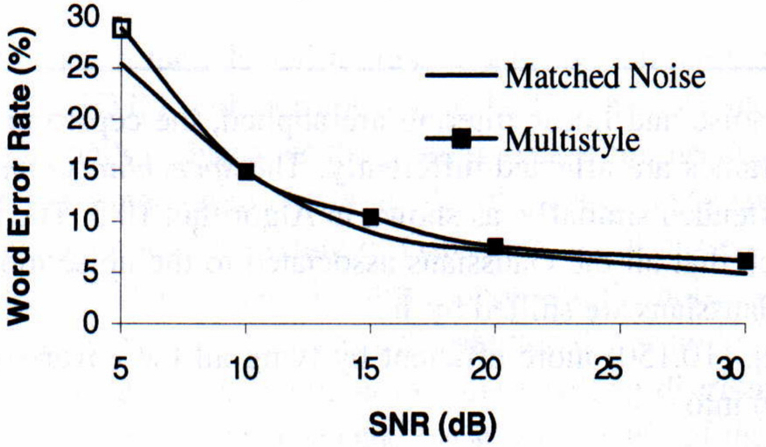


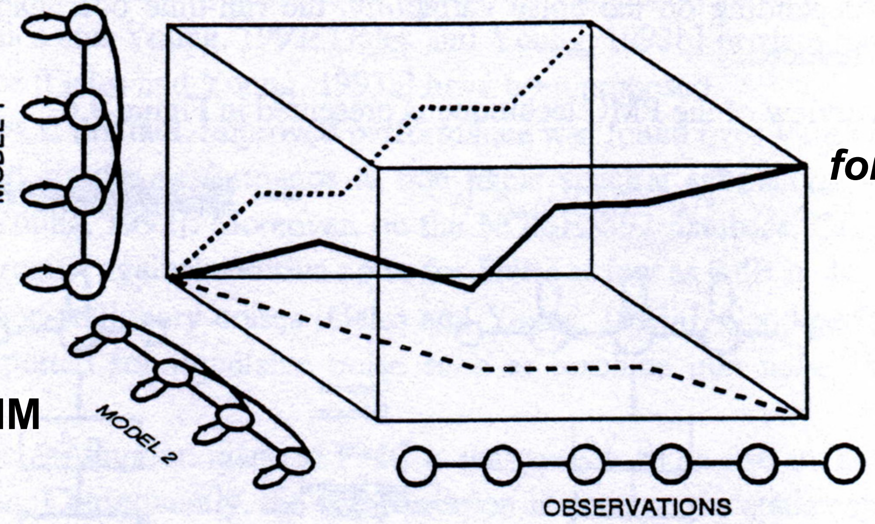
Figure 10.32 Word error rates of multistyle training compared to matched-noise training as a function of the SNR in dB for additive white noise. Whisper is trained as in Figure 10.30. The error rate of multistyle training is between 12% (for low SNR) and 25% (for high SNR) higher in relative terms than that of matched-condition training. Nonetheless, multistyle training does better than a system trained on clean data for all conditions other than clean speech.

Model Adaptation

- A Model-based Noise Compensation Technique
- The standard adaptation methods for speaker adaptation can be used for adapting speech recognizers to noisy environments
  - MAP (Maximum a Posteriori) can offer results similar to those of matched conditions, but it requires a significant amount of adaptation data
  - MLLR (Maximum Likelihood Regression) can achieve reasonable performance with about a minute of speech for minor mismatch. For severe mismatches, MLLR also requires a larger amount of adaptation data

Signal Decomposition Using HMMs

- A Model-based Noise Compensation Technique
- Recognize concurrent signals (speech and noise) simultaneously
  - – Parallel HMMs are used to model the concurrent signals and the composite signal is modeled as a function of their combined outputs
    - Three-dimensional Viterbi Search



Noise HMM
(especially for non-stationary noise)

Clean speech HMM

Computationally Expensive for both Training and Decoding !

FIGURE 9.5 HMM decomposition (after Varga and Moore, 1990).

Parallel Model Combination (PMC)

- A Model-based Noise Compensation Technique
- By using the clean-speech models and a noise model, we can approximate the distributions obtained by training a HMM with corrupted speech

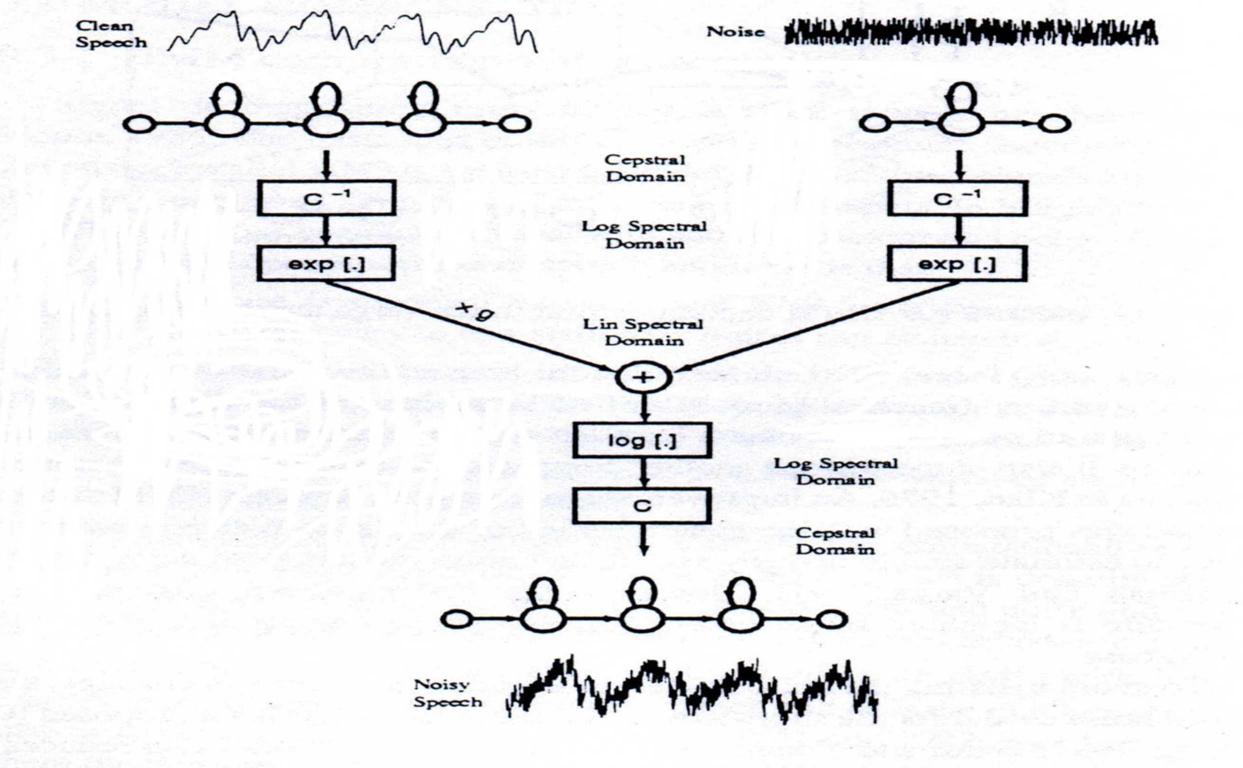
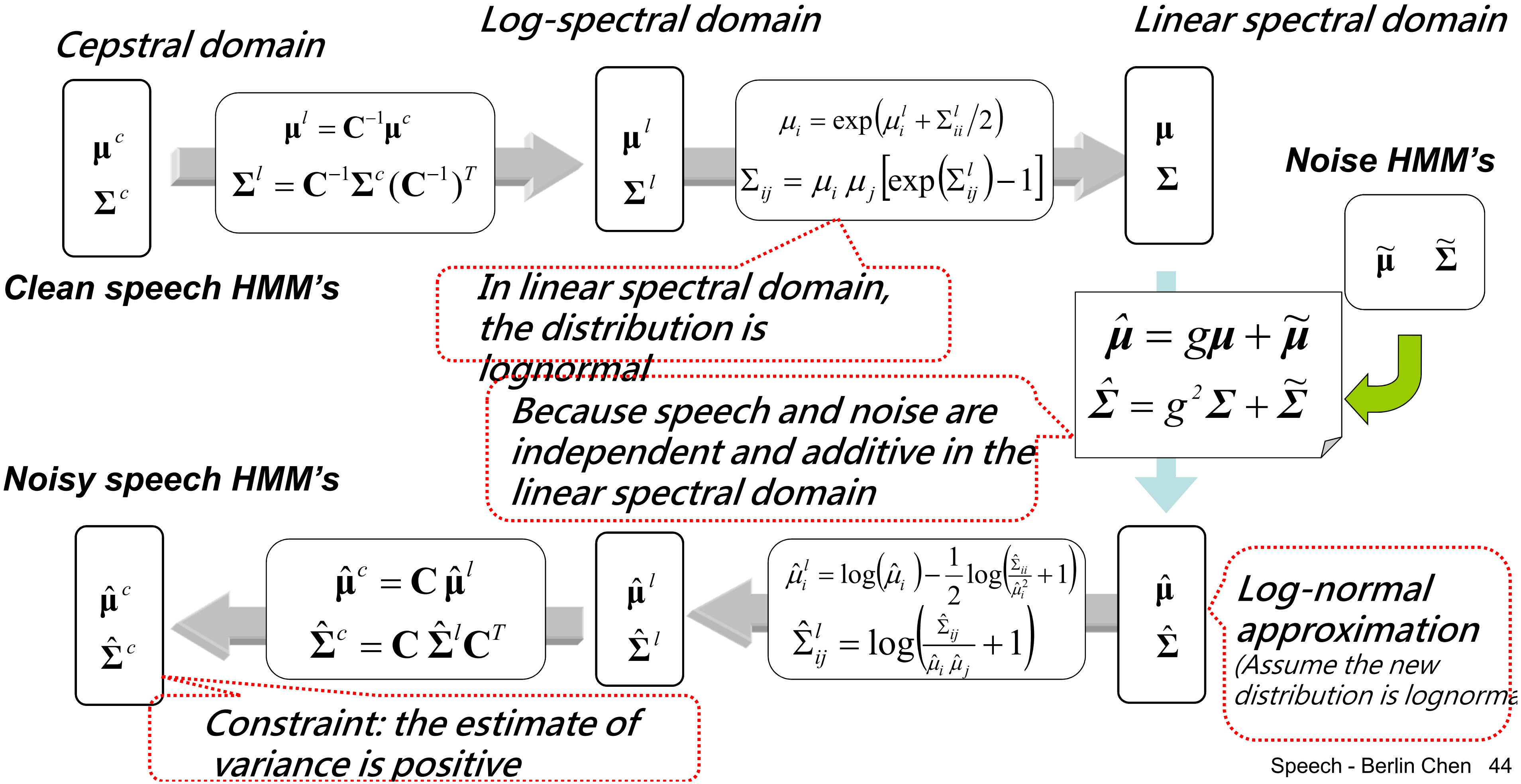


FIGURE 9.6 Principle of Parallel Model Combination (PMC) (after Gales and Young, 1993a). In this figure g is a gain matching term.

Parallel Model Combination (cont.)

- The steps of Standard Parallel Model Combination (Log-Normal Approximation)



Cepstral domain

Log-spectral domain

Linear spectral domain

Clean speech HMM's

$\boldsymbol{\mu}^c$, $\boldsymbol{\Sigma}^c$

$$\boldsymbol{\mu}^l = \mathbf{C}^{-1}\boldsymbol{\mu}^c$$
$$\boldsymbol{\Sigma}^l = \mathbf{C}^{-1}\boldsymbol{\Sigma}^c(\mathbf{C}^{-1})^T$$

$\boldsymbol{\mu}^l$, $\boldsymbol{\Sigma}^l$

$$\mu_i = \exp\left(\mu_i^l + \Sigma_{ii}^l/2\right)$$
$$\Sigma_{ij} = \mu_i\,\mu_j\left[\exp\left(\Sigma_{ij}^l\right) - 1\right]$$

$\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$

In linear spectral domain, the distribution is lognormal

Noise HMM's

$\tilde{\boldsymbol{\mu}}$ $\tilde{\boldsymbol{\Sigma}}$

$$\hat{\boldsymbol{\mu}} = g\boldsymbol{\mu} + \tilde{\boldsymbol{\mu}}$$
$$\hat{\boldsymbol{\Sigma}} = g^2\boldsymbol{\Sigma} + \tilde{\boldsymbol{\Sigma}}$$

Because speech and noise are independent and additive in the linear spectral domain

Noisy speech HMM's

$\hat{\boldsymbol{\mu}}$, $\hat{\boldsymbol{\Sigma}}$

Log-normal approximation
(Assume the new distribution is lognorma

$$\hat{\mu}_i^l = \log\left(\hat{\mu}_i\right) - \frac{1}{2}\log\left(\frac{\hat{\Sigma}_{ii}}{\hat{\mu}_i^2} + 1\right)$$
$$\hat{\Sigma}_{ij}^l = \log\left(\frac{\hat{\Sigma}_{ij}}{\hat{\mu}_i\,\hat{\mu}_j} + 1\right)$$

$\hat{\boldsymbol{\mu}}^l$, $\hat{\boldsymbol{\Sigma}}^l$

$$\hat{\boldsymbol{\mu}}^c = \mathbf{C}\,\hat{\boldsymbol{\mu}}^l$$
$$\hat{\boldsymbol{\Sigma}}^c = \mathbf{C}\,\hat{\boldsymbol{\Sigma}}^l\mathbf{C}^T$$

$\hat{\boldsymbol{\mu}}^c$, $\hat{\boldsymbol{\Sigma}}^c$

Constraint: the estimate of variance is positive

Parallel Model Combination (cont.)

- Modification-I: Perform the model combination in the Log-Spectral Domain (the simplest approximation)
  - Log-Add Approximation: (without compensation of variances)

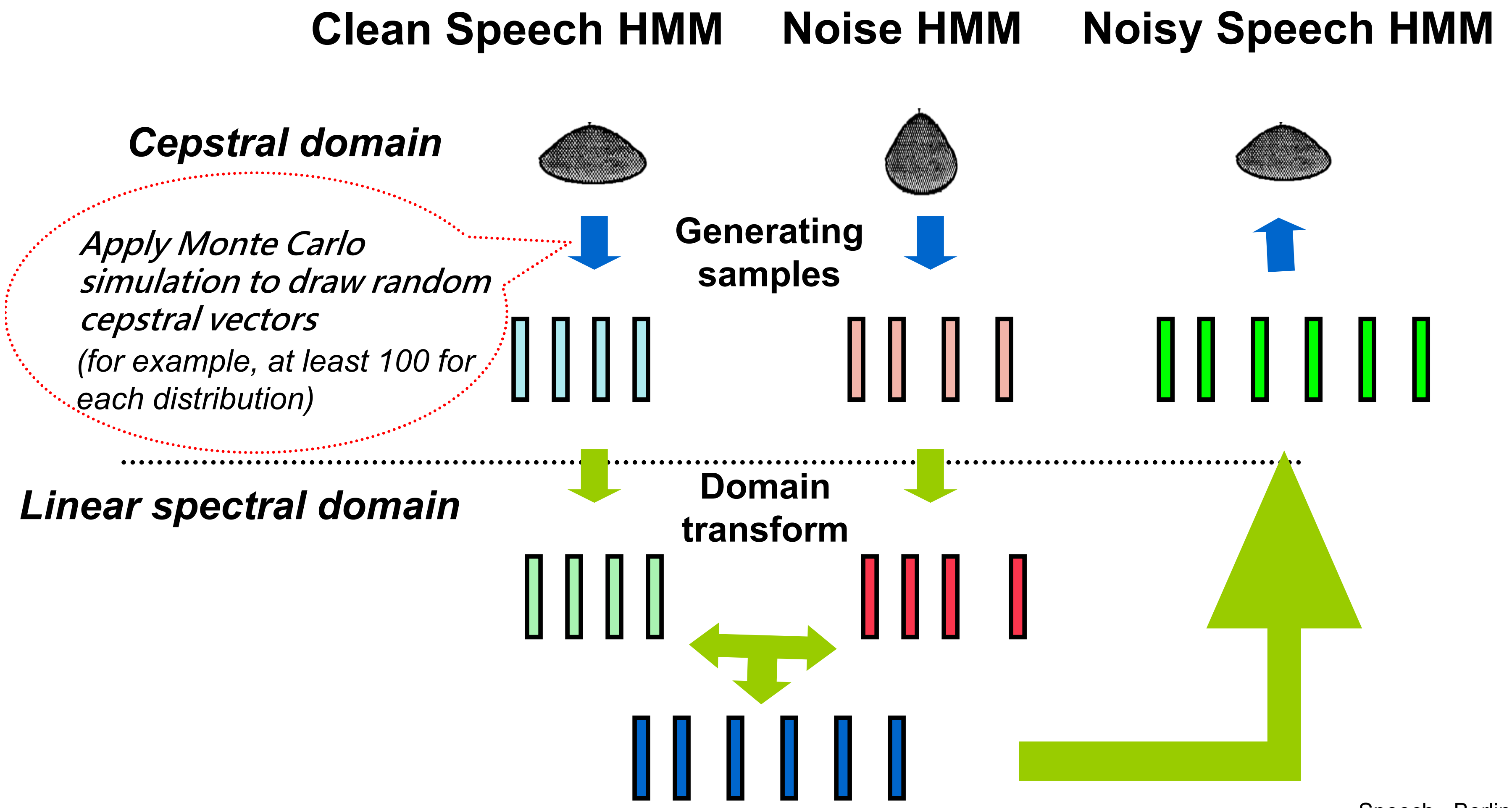
$$\hat{\mu}^l = \log\left(\exp\left(\mu^l\right) + \exp\left(\tilde{\mu}^l\right)\right)$$

    - The variances are assumed to be small
  - A simplified version of Log-Normal approximation
    - Reduction in computational load

- Modification-II: Perform the model combination in the Linear Spectral Domain (Data-Driven PMC, DPMC, or Iterative PMC)
  - Use the speech models to generate noisy samples (corrupted speech observations) and then compute a maximum likelihood of these noisy samples
  - This method is less computationally expensive than standard PMC with comparable performance

Parallel Model Combination (cont.)

- Modification-II: Perform the model combination in the Linear Spectral Domain (Data-Driven PMC, DPMC)



Parallel Model Combination (cont.)

- Data-Driven PMC

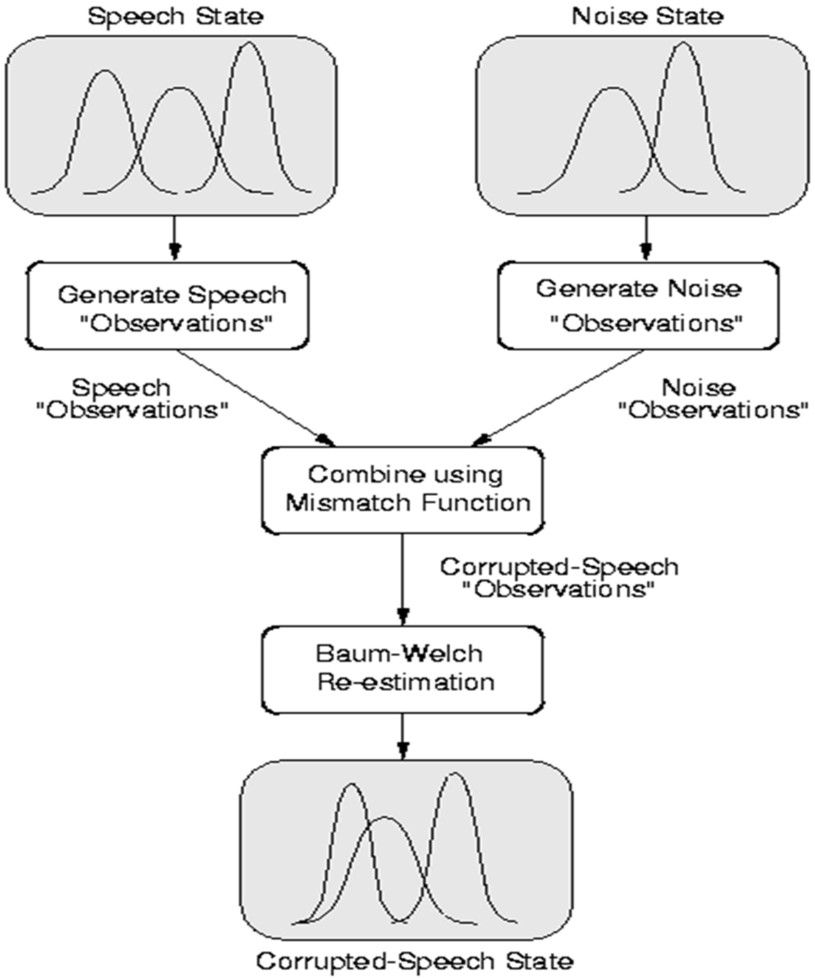

Speech State
Noise State
Generate Speech "Observations"
Generate Noise "Observations"
Speech "Observations"
Noise "Observations"
Combine using Mismatch Function
Corrupted-Speech "Observations"
Baum-Welch Re-estimation
Corrupted-Speech State

Figure 5.3: Data-driven parallel model combination

Vector Taylor Series (VTS)

P. J. Moreno,1995

- **A Model-based Noise Compensation Technique**
- VTS Approach
  - Similar to PMC, the noisy-speech-like models is generated by combining of clean speech HMM's and the noise HMM
  - Unlike PMC, the VTS approach combines the parameters of clean speech HMM's and the noise HMM *linearly* in the *log-spectral domain*

$$P_X(\omega)=P_S(\omega)P_H(\omega)+P_N(\omega)$$

Power spectrum

$$X^l=\log\left(P_S(\omega)P_H(\omega)+P_N(\omega)\right)$$

Log Power spectrum

$$=\log\left(P_S(\omega)P_H(\omega)\left(1+\frac{P_N(\omega)}{P_S(\omega)P_H(\omega)}\right)\right)$$

$$=\log P_S(\omega)+\log P_H(\omega)+\log\left(1+e^{\log P_N(\omega)-\log P_S(\omega)-\log P_H(\omega)}\right)$$

$$=S^l+H^l+\log\left(1+e^{N^l-S^l-H^l}\right)$$

Non-linear function

$$=S^l+H^l+f\left(S^l,H^l,N^l\right),\quad \text{where} f\left(S^l,H^l,N^l\right)=\log\left(1+e^{N^l-S^l-H^l}\right)$$

Is a vector function

Vector Taylor Series (cont.)

- The Taylor series provides a polynomial representation of a function in terms of the function and its derivatives at a point
  - Application often arises when nonlinear functions are employed and we desire to obtain a linear approximation
  - The function is represented as an offset and a linear term

$$f : R \rightarrow R$$

$$f(x) = \underline{f(x_0) + f'(x_0)(x - x_0)} + \frac{1}{2} f''(x_0)(x - x_0)^2$$

$$+ \ldots + \frac{1}{n!} f^{(n)}(x_0)(x - x_0)^n + o\left(\left|x - x_0\right|^n\right)$$

Vector Taylor Series (cont.)

- Apply Taylor Series Approximation

$$f\left(N^l, S^l, H^l\right) \cong f\left(S_0{}^l, H_0{}^l, N_0{}^l\right) + \frac{df\left(S_0{}^l, H_0{}^l, N_0{}^l\right)}{dS^l}\left(S^l - S_0{}^l\right)$$

$$+ \frac{df\left(S_0{}^l, H_0{}^l, N_0{}^l\right)}{dH^l}\left(H^l - H_0{}^l\right) + \frac{df\left(S_0{}^l, H_0{}^l, N_0{}^l\right)}{dN^l}\left(N^l - N_0{}^l\right) + ...$$

  – VTS-0: use only the 0th-order terms of Taylor Series
  – VTS-1: use only the 0th- and 1th-order terms of Taylor Series
  – $f\left(S_0{}^l, H_0{}^l, N_0{}^l\right)$ is the vector function evaluated at a particular vector point

- **If VTS-0 is used**

$$E\left[X^l\right] = E\left[S^l + H^l + f\left(S^l, H^l, N^l\right)\right]$$

$$u_x^l = u_s^l + u_h^l + E\left[f\left(S^l, H^l, N^l\right)\right]$$

0-th order VTS

$$\cong u_s^l + u_h^l + E\left[f\left(u_s^l, u_h^l, u_n^l\right)\right]$$

$$\cong u_s^l + u_h^l + f\left(u_s^l, u_h^l, u_n^l\right) \quad (X^l \text{ is also Gaussian})$$

$$\Sigma_x^l \cong \Sigma_s^l + \Sigma_h^l \quad (\text{if } S^l \text{ and } H^l \text{ are independent})$$

To get the clean speech statistics

If the channel filter is linear - time invariant, we can regard it as a bias (constant) , g, in the log power spectrum domain

$$u_x^l \cong u_s^l + g + f\left(u_s^l, g, u_n^l\right) \quad (X^l \text{ is also Gaussian})$$

$$\Sigma_x^l \cong \Sigma_s^l$$

Vector Taylor Series (cont.)

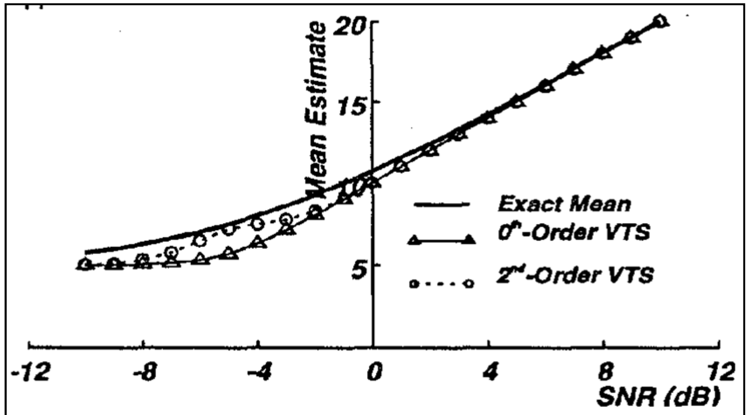


Figure 1. Effects of noise on the mean of the incoming signal. The exact values of the mean and estimates of the mean obtained from the zeroth-order and second-order VTS expansion are compared over a range of SNRs.

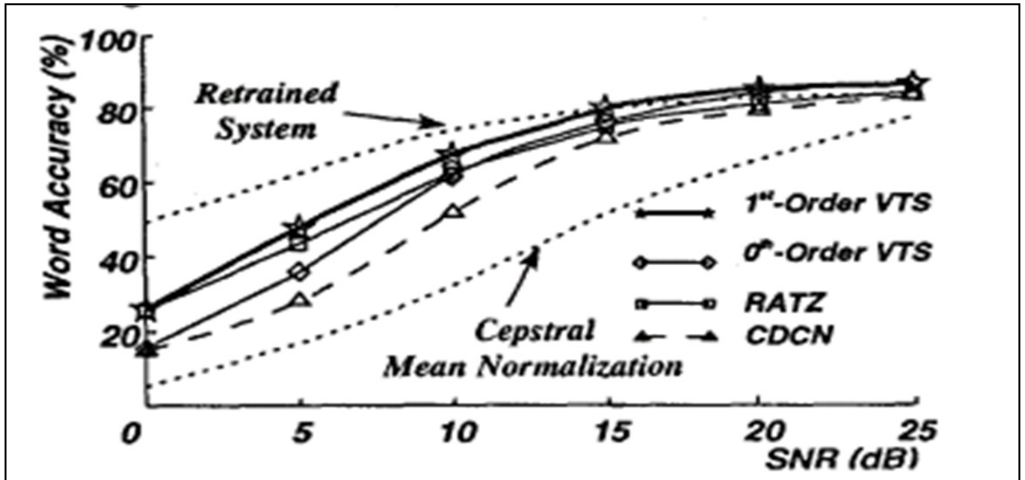


Figure 3. Comparison of recognition accuracy obtained for th CENSUS database using the zeroth-order and first-order VTS, CDCN, and RATZ algorithms as a function of SNR. The dotte curves indicate baseline performance using cepstral mean normalization only, as well as results obtained by completely retraining the system in the new environment.

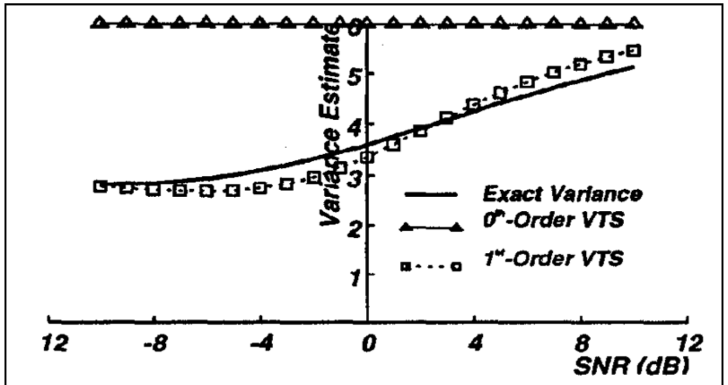


Figure 2. Effects of noise on the variance of the signal. The exact values of the variance and estimates of the variance obtained from the zeroth-order and first-order VTS expansion are compared over a range of SNRs.

Retraining on Compensated Features

- A Model-based Noise Compensation Technique that also Uses enhanced Features (processed by SS, CMN, etc.)
  - Combine speech enhancement and model compensation

SPLICE: Stereo-based Piecewise Linear Compensation

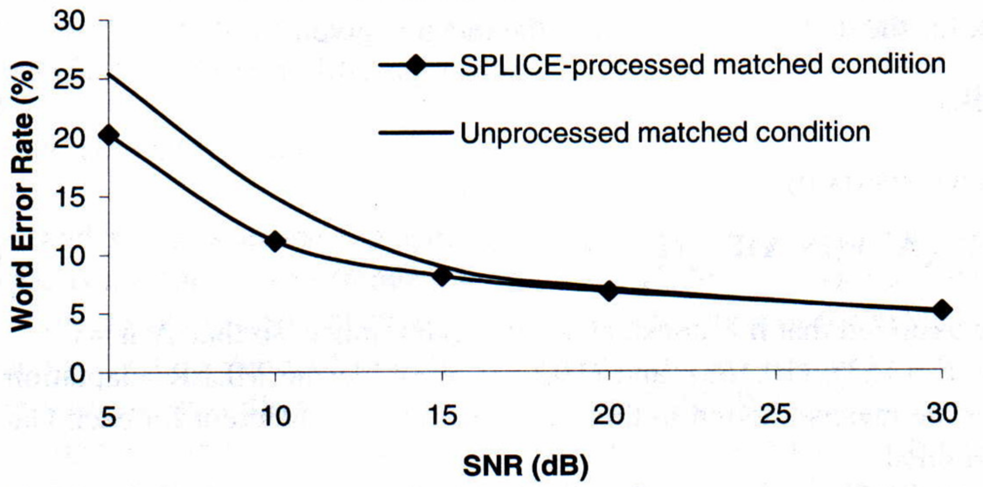
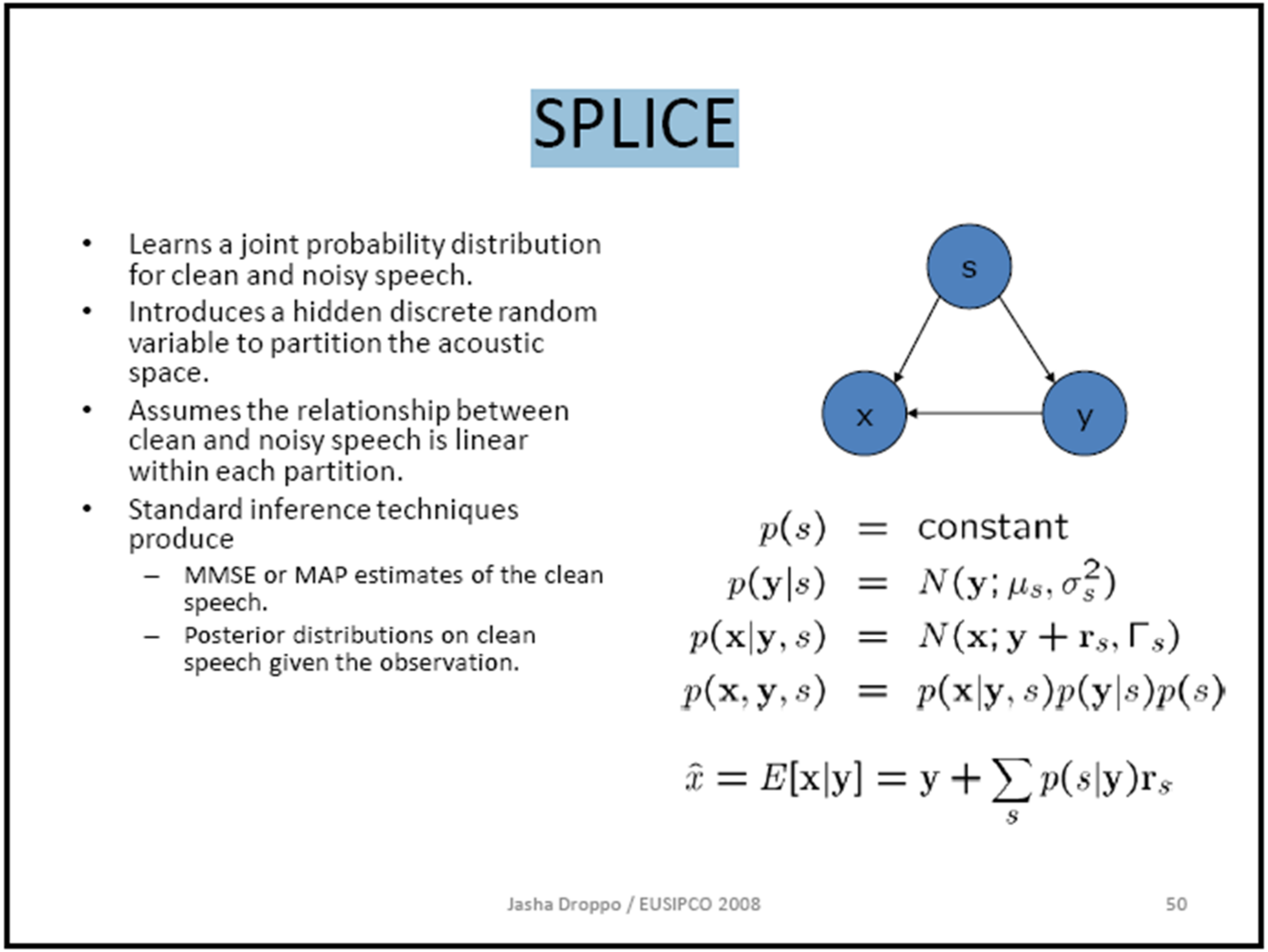


Figure 10.36 Word error rates of matched-noise training without feature preprocessing and with the SPLICE algorithm [21] as a function of the SNR in dB for additive white noise. Whisper is trained as in Figure 10.30. Error rate with the mixture Gaussian model is up to 30% lower than that of standard noisy matched conditions for low SNRs while it is about the same for high SNRs.

More on SPLICE



SPLICE

- Learns a joint probability distribution for clean and noisy speech.
- Introduces a hidden discrete random variable to partition the acoustic space.
- Assumes the relationship between clean and noisy speech is linear within each partition.
- Standard inference techniques produce
  - MMSE or MAP estimates of the clean speech.
  - Posterior distributions on clean speech given the observation.

$$
\begin{aligned}
p(s) &= \text{constant} \\
p(\mathbf{y}|s) &= N(\mathbf{y}; \mu_s, \sigma_s^2) \\
p(\mathbf{x}|\mathbf{y}, s) &= N(\mathbf{x}; \mathbf{y} + \mathbf{r}_s, \Gamma_s) \\
p(\mathbf{x}, \mathbf{y}, s) &= p(\mathbf{x}|\mathbf{y}, s)p(\mathbf{y}|s)p(s)
\end{aligned}
$$

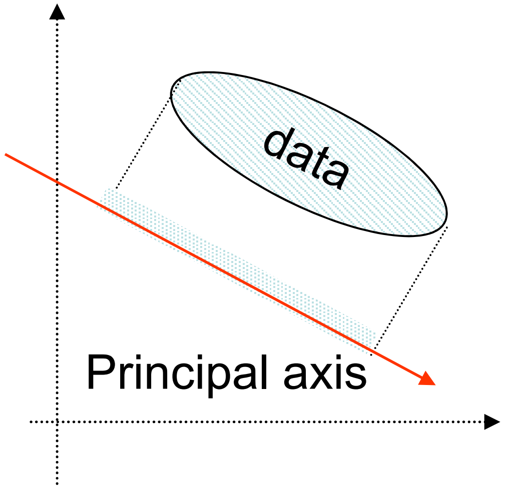
$$\hat{x} = E[\mathrm{x}|\mathrm{y}] = \mathrm{y} + \sum_s p(s|\mathrm{y})\mathbf{r}_s$$

Jasha Droppo / EUSIPCO 2008 50

Note that this slide was adapted from Dr. Droppo’s presentation slides

Principal Component Analysis

- Principal Component Analysis (PCA) :
  - Widely applied for the data analysis and dimensionality reduction in order to derive the most “expressive” feature
  - Criterion:
    for **a zero mean** r.v. $\boldsymbol{x} \in \boldsymbol{R}^N$, find k ($k \leq N$) **orthonormal vectors** $\{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_k\}$ so that
  - (1) $var(\mathbf{e}_1^T \boldsymbol{x}) = max\ 1$
    (2) $var(\mathbf{e}_i^T \boldsymbol{x}) = max\ i$
    subject to $\mathbf{e}_i \perp \mathbf{e}_{i-1} \perp \ldots\ldots \perp \mathbf{e}_1 \quad 1 \leq i \leq k$
  - $\{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_k\}$ are in fact the **eigenvectors** of **the covariance matrix ($\Sigma_{\boldsymbol{x}}$) for $\boldsymbol{x}$** corresponding to the largest k eigenvalues
  - Final r.v $\boldsymbol{y} \in \boldsymbol{R}^k$ **:** the linear transform (projection) of the original r.v., $\boldsymbol{y} = \boldsymbol{A}^T \boldsymbol{x}$
    $\boldsymbol{A} = [\mathbf{e}_1\ \mathbf{e}_2 \ldots\ldots\ \mathbf{e}_k]$



Principal Component Analysis (cont.)

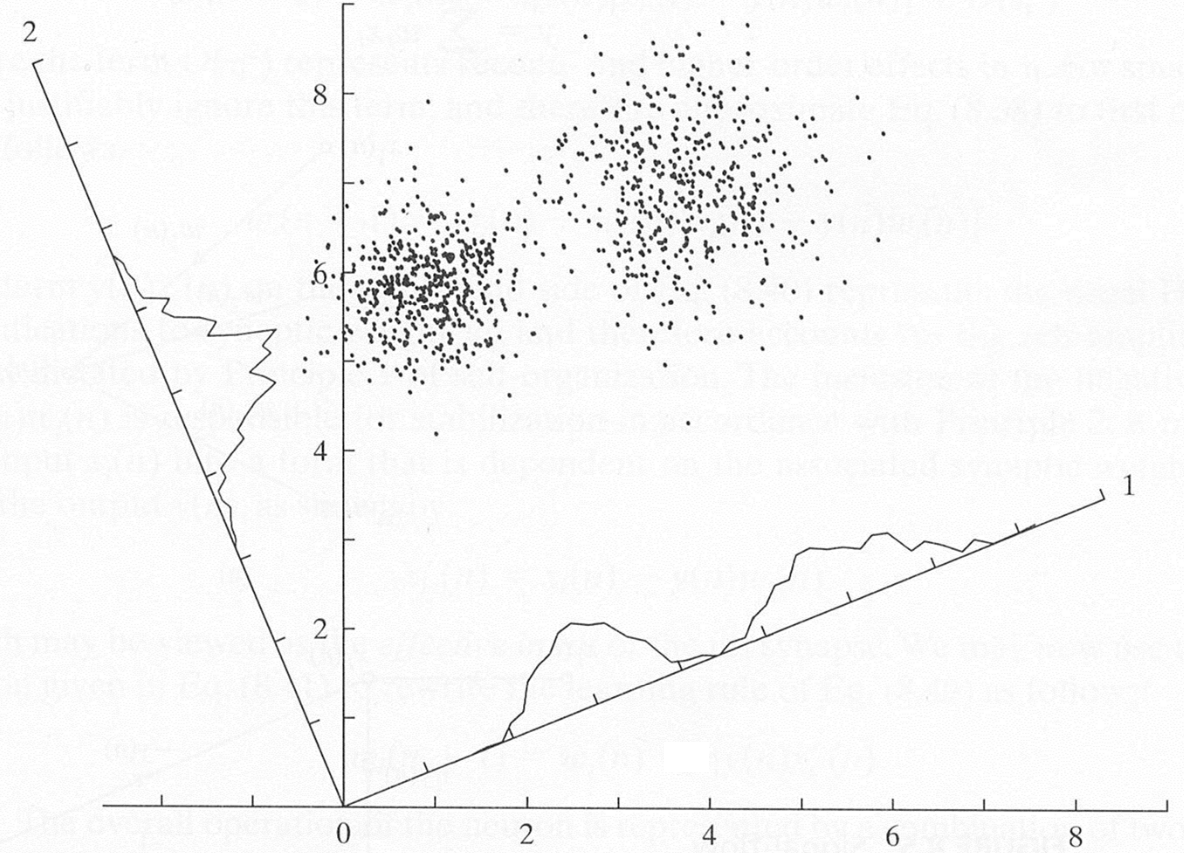


FIGURE 8.4 A cloud of data points is shown in two dimensions, and the density plots formed by projecting this cloud onto each of two axes, 1 and 2, are indicated. The projection onto axis 1 has maximum variance, and clearly shows the bimodal, or clustered character of the data.

Principal Component Analysis (cont.)

- Properties of PCA
  - The components of ***y*** are mutually uncorrelated

    $E\{y_i y_j\}=E\{(\mathbf{e}_i^T\boldsymbol{x})(\mathbf{e}_j^T\boldsymbol{x})^T\}=E\{(\mathbf{e}_i^T\boldsymbol{x})(\boldsymbol{x}^T\mathbf{e}_j)\}=\mathbf{e}_i^T E\{\boldsymbol{x}\boldsymbol{x}^T\}\,\mathbf{e}_j=\mathbf{e}_i^T\Sigma_{\boldsymbol{x}}\mathbf{e}_j$
    $=\lambda_j\mathbf{e}_i^T\mathbf{e}_j=0$, *if* $i\neq j$

    $\therefore$ **the covariance of *y* is diagonal**

  - The error power (mean-squared error) between the original vector ***x*** and the projected ***x***' is minimum

    $\boldsymbol{x}=(\mathbf{e}_1^T\boldsymbol{x})\mathbf{e}_1+(\mathbf{e}_2^T\boldsymbol{x})\mathbf{e}_2+\ldots\ldots+(\mathbf{e}_k^T\boldsymbol{x})\mathbf{e}_k+\ldots\ldots+(\mathbf{e}_N^T\boldsymbol{x})\mathbf{e}_N$

    $\boldsymbol{x}'=(\mathbf{e}_1^T\boldsymbol{x})\mathbf{e}_1+(\mathbf{e}_2^T\boldsymbol{x})\mathbf{e}_2+\ldots\ldots+(\mathbf{e}_k^T\boldsymbol{x})\mathbf{e}_k$ *(Note : $\boldsymbol{x}'\in\boldsymbol{R}^N$)*

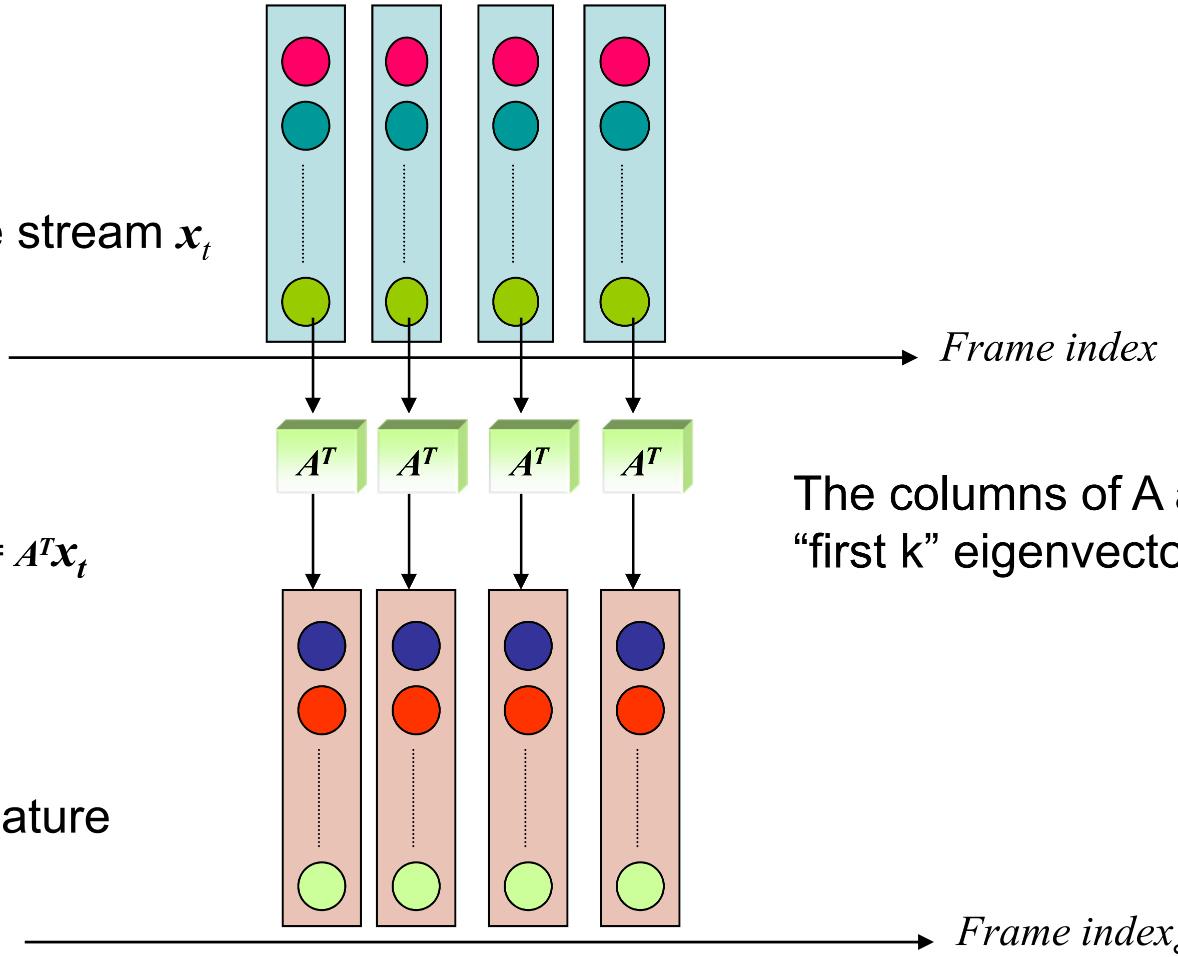
    error r.v :

    $\boldsymbol{x}-\boldsymbol{x}'=(\mathbf{e}_{k+1}^T\boldsymbol{x})\mathbf{e}_{k+1}+(\mathbf{e}_{k+2}^T\boldsymbol{x})\mathbf{e}_{k+2}+\ldots\ldots+(\mathbf{e}_N^T\boldsymbol{x})\mathbf{e}_N$

    $E((\boldsymbol{x}-\boldsymbol{x}')^T(\boldsymbol{x}-\boldsymbol{x}'))=E((\mathbf{e}_{k+1}^T\boldsymbol{x})\,\mathbf{e}_{k+1}^T\mathbf{e}_{k+1}(\mathbf{e}_{k+1}^T\boldsymbol{x}))+\ldots\ldots+E((\mathbf{e}_N^T\boldsymbol{x})\,\mathbf{e}_N^T\mathbf{e}_N(\mathbf{e}_N^T\boldsymbol{x}))$
    $=\mathrm{var}(\mathbf{e}_{k+1}^T\boldsymbol{x})+\mathrm{var}(\mathbf{e}_{k+2}^T\boldsymbol{x})+\ldots\ldots\ \mathrm{var}(\mathbf{e}_N^T\boldsymbol{x})$
    $=\lambda_{k+1}+\lambda_{k+2}+\ldots\ldots+\lambda_N$ ➔ minimum

PCA Applied in Inherently Robust Features

- Application 1 : **the linear transform of the original** features (in the spatial domain)



Original feature stream $\boldsymbol{x}_t$

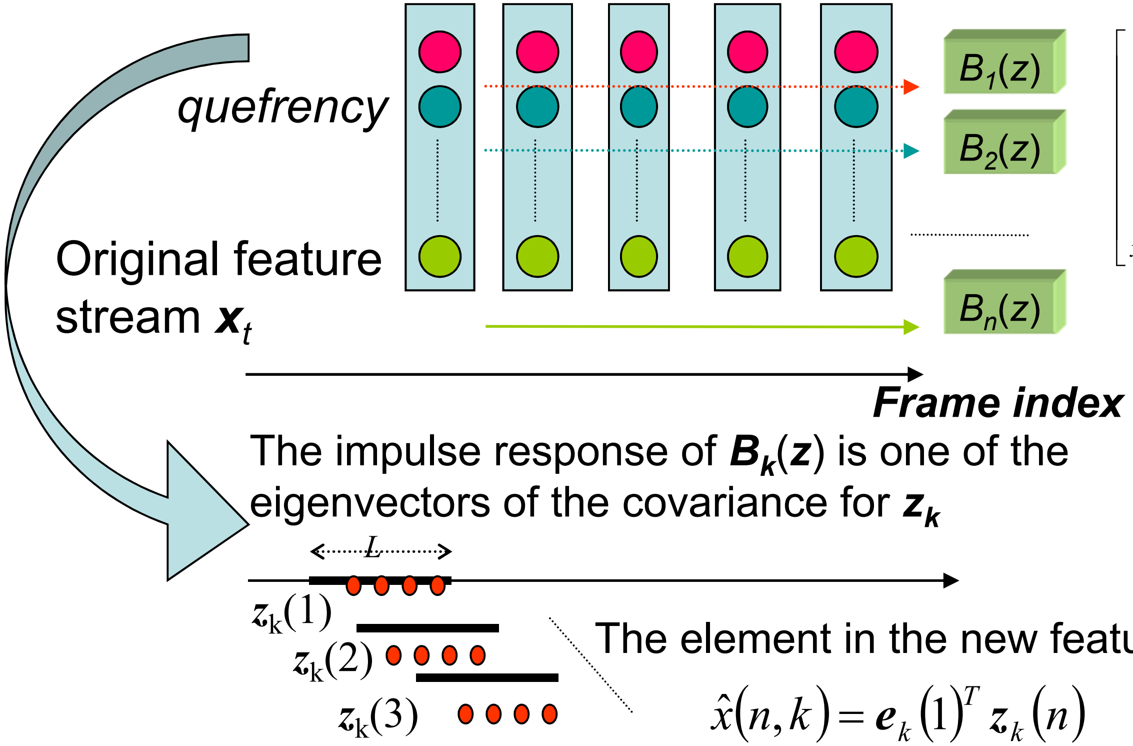
$z_t = A^T \boldsymbol{x}_t$

The columns of A are the "first k" eigenvectors of $\Sigma_{\mathbf{x}}$

transformed feature stream z_t

PCA Applied in Inherently Robust Features (cont.)

- Application 2 : **PCA-derived temporal filter** (in the temporal domain)
  - The effect of the temporal filter is equivalent to the weighted sum of sequence of a specific MFCC coefficient with length L slid along the frame index



The impulse response of $\boldsymbol{B_k(z)}$ is one of the eigenvectors of the covariance for $\boldsymbol{z_k}$

$$\underset{\mathbf{x}(1)}{\begin{bmatrix} x(1,1)\\ x(1,2)\\ \vdots\\ x(1,k)\\ \vdots\\ x(1,K)\end{bmatrix}}\underset{\mathbf{x}(2)}{\begin{bmatrix} x(2,1)\\ x(2,2)\\ \vdots\\ x(2,k)\\ \vdots\\ x(2,K)\end{bmatrix}}\underset{\mathbf{x}(3)}{\begin{bmatrix} x(3,1)\\ x(3,2)\\ \vdots\\ x(3,k)\\ \vdots\\ x(3,K)\end{bmatrix}}\underset{\cdots}{\cdots}\underset{\mathbf{x}(n)}{\begin{bmatrix} x(n,1)\\ x(n,2)\\ \vdots\\ x(n,k)\\ \vdots\\ x(n,K)\end{bmatrix}}\underset{\cdots}{\cdots}\underset{\mathbf{x}(N)}{\begin{bmatrix} x(N,1)\\ x(N,2)\\ \vdots\\ x(N,k)\\ \vdots\\ x(N,K)\end{bmatrix}}\begin{matrix}\to y_1(m)\\ \to y_2(m)\\ \vdots\\ \to y_k(m)\\ \vdots\\ \to y_K(m)\end{matrix}$$

$$\mathbf{z_k}(n)=[\, y_k(n)\ y_k(n+1)\ y_k(n+2)\ \ldots\ldots\ y_k(n+L\text{-}1)]^{\mathrm{T}}$$

$$\mu_{z_k} = \frac{1}{N-L+1}\sum_{n=1}^{N-L+1} z_k(n)$$

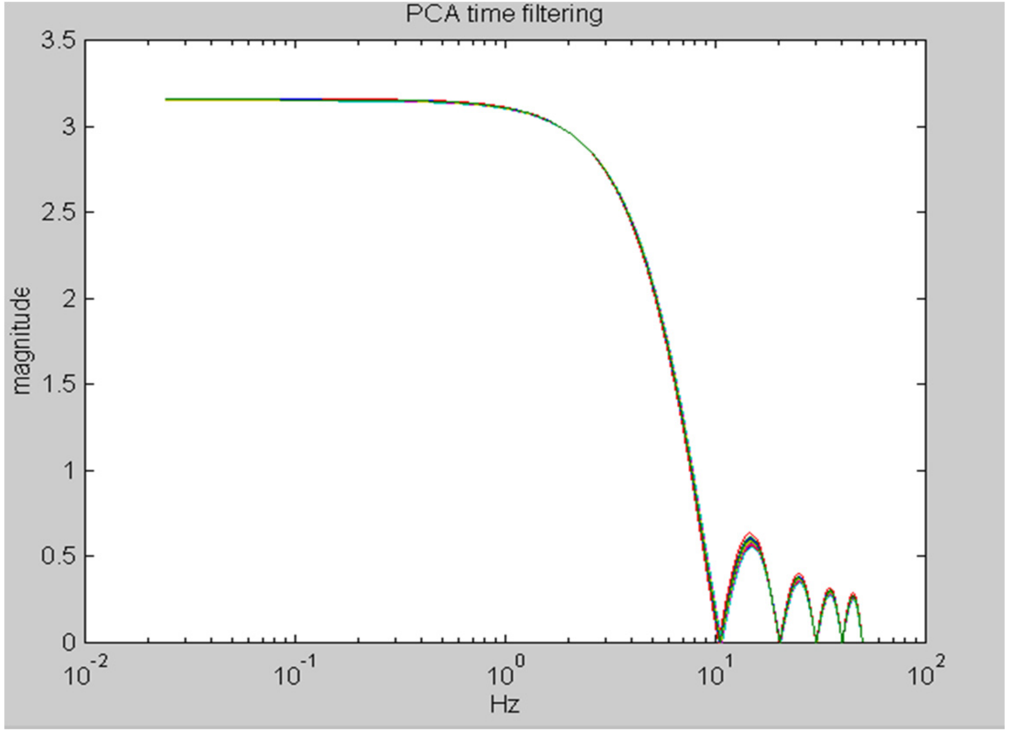
$$\Sigma_{z_k} = \frac{1}{N-L+1}\sum_{n=1}^{N-L+1}\left(z_k(n)-\mu_{z_k}\right)\left(z_k(n)-\mu_{z_k}\right)^T$$

The element in the new feature vector

$$\hat{x}(n,k) = \boldsymbol{e}_k(1)^T\, \boldsymbol{z}_k(n)$$

From Dr. Jei-wei Hung

PCA Applied in Inherently Robust Features (cont.)



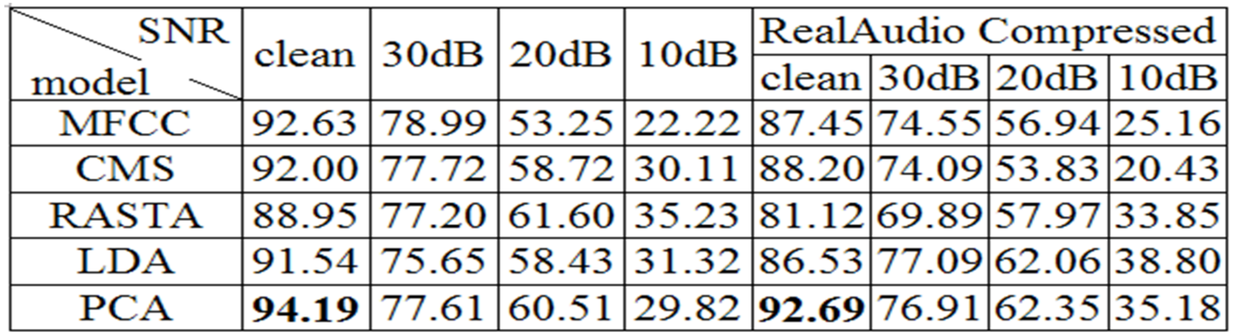
The frequency responses of the 15 PCA-derived temporal filters

From Dr. Jei-wei Hung

PCA Applied in Inherently Robust Features (cont.)

- Application 2 : **PCA-derived temporal filter**

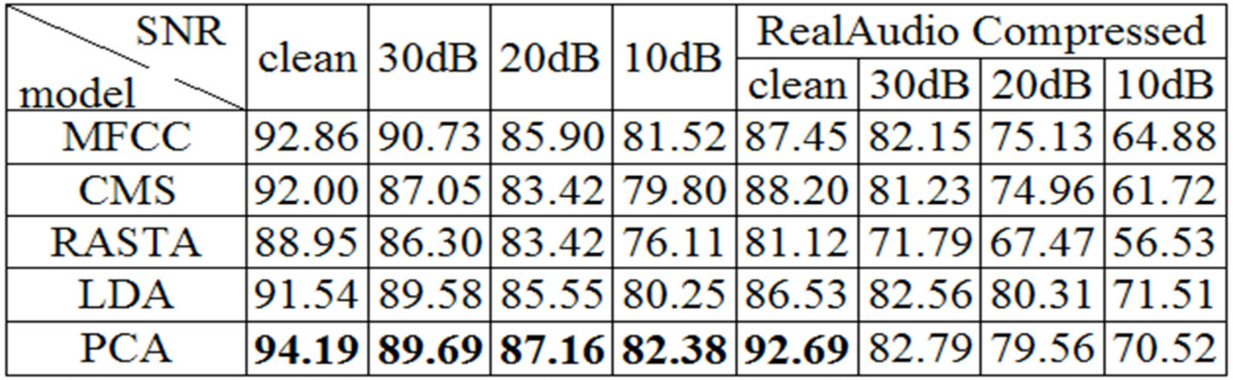
Filter length L=10



SNR / model	clean	30dB	20dB	10dB	RealAudio Compressed			
					clean	30dB	20dB	10dB
MFCC	92.63	78.99	53.25	22.22	87.45	74.55	56.94	25.16
CMS	92.00	77.72	58.72	30.11	88.20	74.09	53.83	20.43
RASTA	88.95	77.20	61.60	35.23	81.12	69.89	57.97	33.85
LDA	91.54	75.65	58.43	31.32	86.53	77.09	62.06	38.80
PCA	**94.19**	77.61	60.51	29.82	**92.69**	76.91	62.35	35.18

Table 1: The digit recognition rates for different versions of HMM's with 5 states and 4 mixtures per state under mismatched conditions

Mismatched condition



SNR / model	clean	30dB	20dB	10dB	RealAudio Compressed			
					clean	30dB	20dB	10dB
MFCC	92.86	90.73	85.90	81.52	87.45	82.15	75.13	64.88
CMS	92.00	87.05	83.42	79.80	88.20	81.23	74.96	61.72
RASTA	88.95	86.30	83.42	76.11	81.12	71.79	67.47	56.53
LDA	91.54	89.58	85.55	80.25	86.53	82.56	80.31	71.51
PCA	**94.19**	**89.69**	**87.16**	**82.38**	**92.69**	82.79	79.56	70.52

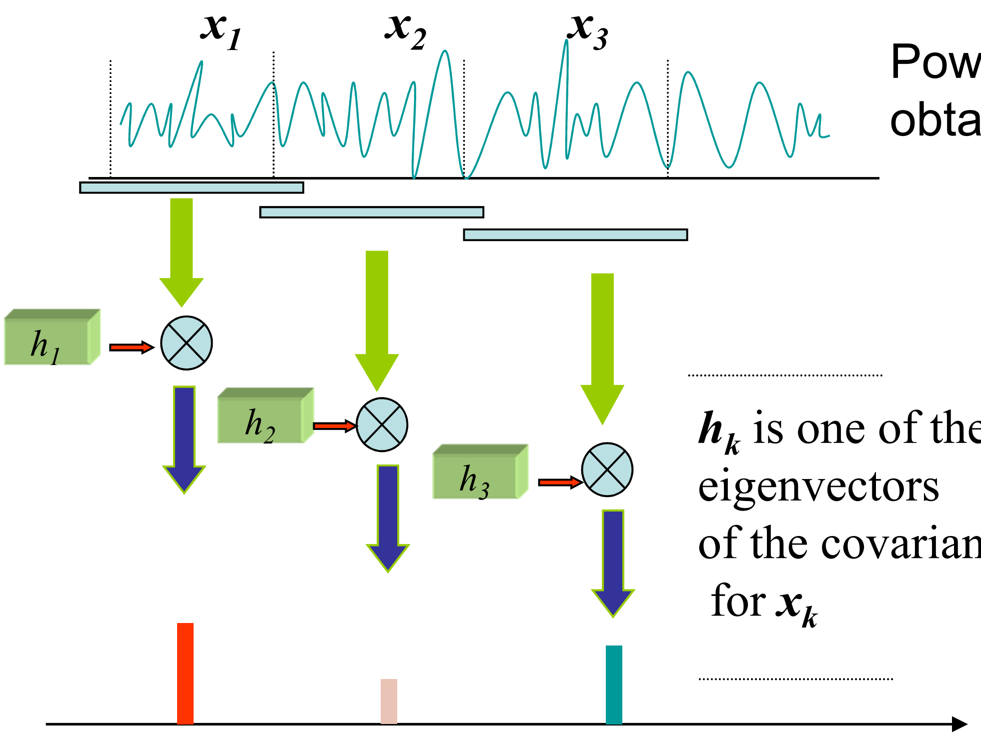
Table 2: The digit recognition rates for different versions of HMM's with 5 states and 4 mixtures per state under matched noisy conditions

Matched condition

From Dr. Jei-wei Hung

PCA Applied in Inherently Robust Features (cont.)

- Application 3 : **PCA-derived filter bank**



Power spectrum
obtained by DFT

h_k is one of the eigenvectors
of the covariance
for x_k

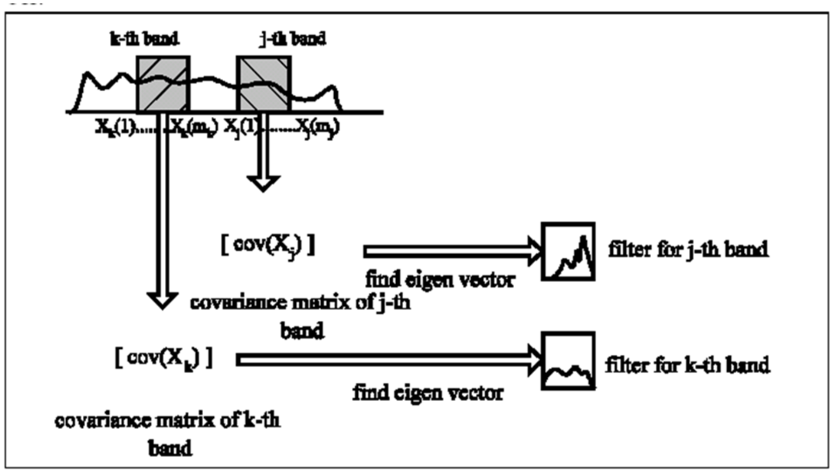


Figure 1: The process of finding PCA-optimized filter bank coefficients

From Dr. Jei-wei Hung

PCA Applied in Inherently Robust Features (cont.)

- Application 3 : **PCA-derived filter bank**

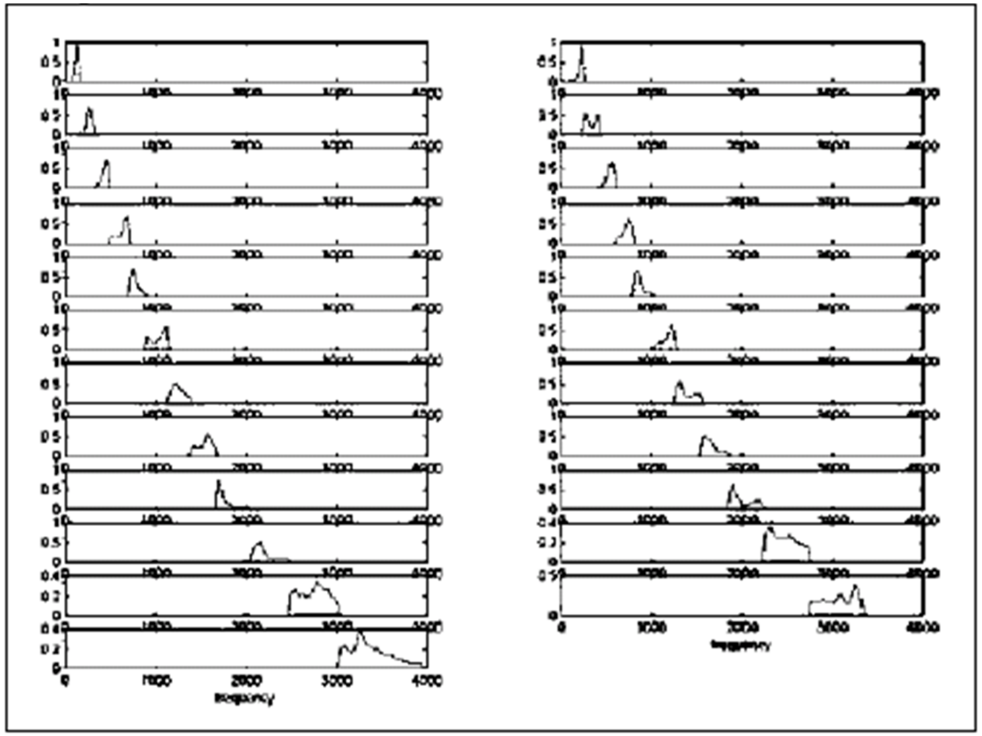


Figure 2: The shape of 23 filters in the filter-bank

From Dr. Jei-wei Hung

Linear Discriminative Analysis

- Linear Discriminative Analysis (LDA)
  - Widely applied for the pattern classification
  - In order to derive the most "discriminative" feature
  - ***Criterion*** : assume $\boldsymbol{w}_j$, μ_j and Σ_j are the weight, mean and covariance of class j, $j=1 \ldots\ldots N$. Two matrices are defined as:

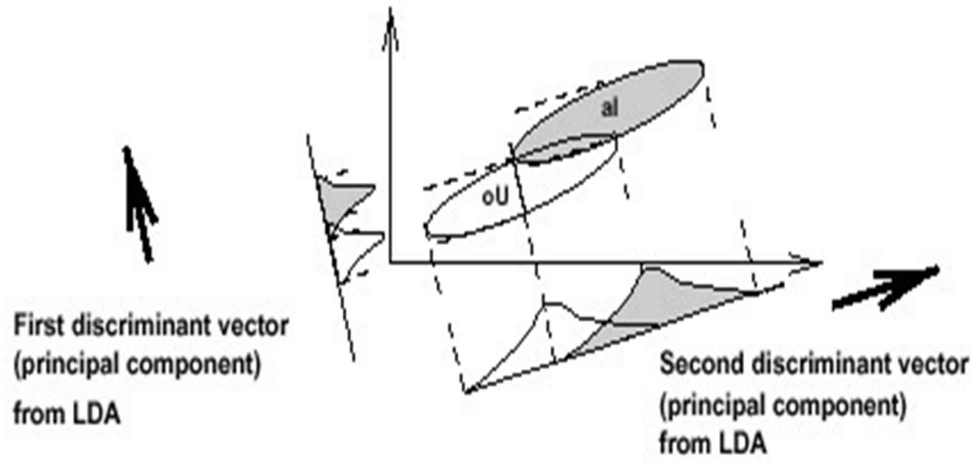
$$\text{Between-class covariance}: \boldsymbol{S}_b = \sum_{j=1}^{N} w_j \left(\boldsymbol{\mu}_j - \boldsymbol{\mu}\right)\left(\boldsymbol{\mu}_j - \boldsymbol{\mu}\right)^T$$

$$\text{Within-class covariance}: \boldsymbol{S}_w = \sum_{j=1}^{N} w_j \boldsymbol{\Sigma}_j$$

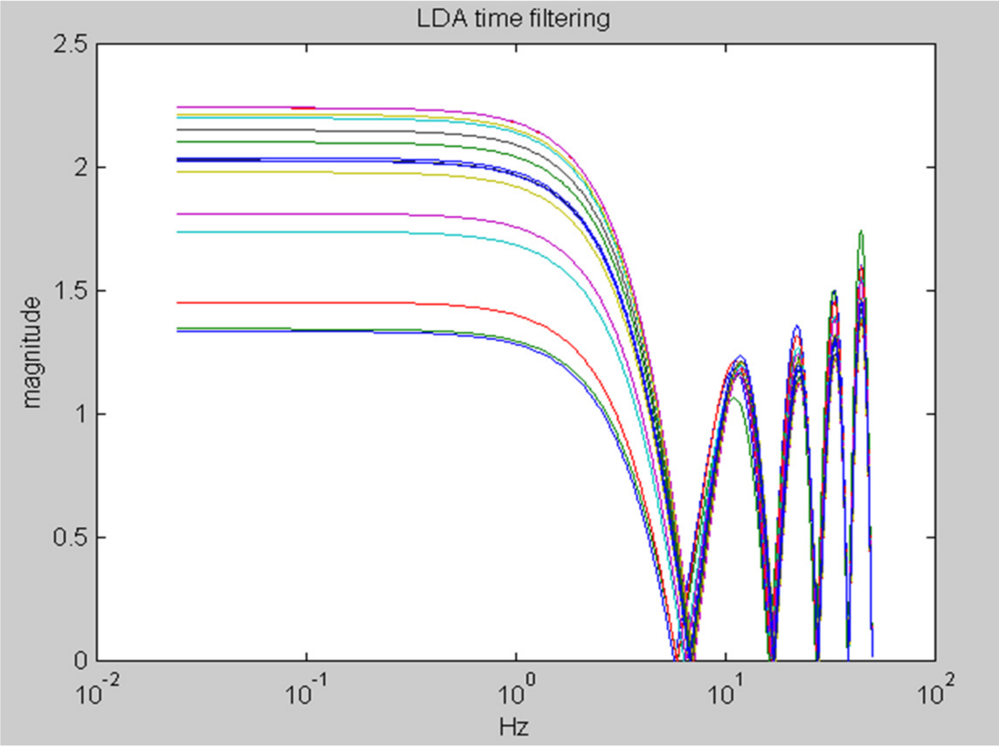
Find $\mathbf{W}=[\boldsymbol{w}_1\ \boldsymbol{w}_2 \ldots\ldots \boldsymbol{w}_k]$ such that

$$\hat{\boldsymbol{W}} = \arg\max_{\boldsymbol{W}} \frac{\left|\boldsymbol{W}^T \boldsymbol{S}_b \boldsymbol{W}\right|}{\left|\boldsymbol{W}^T \boldsymbol{S}_w \boldsymbol{W}\right|}$$

  - The columns $\boldsymbol{w}_j$ of $\mathbf{W}$ are the eigenvectors of $\mathbf{S}_w^{-1}\mathbf{S}_B$ having the largest eigenvalues


al
oU
First discriminant vector (principal component) from LDA
Second discriminant vector (principal component) from LDA

Linear Discriminative Analysis (cont.)



The frequency responses of the 15 LDA-derived temporal filters

From Dr. Jei-wei Hung

Minimum Classification Error

- Minimum Classification Error (MCE):
  - General Objective : find an optimal feature presentation or an optimal recognition model to ***minimize the expected error of classification***
  - The recognizer is often operated under the following decision rule :

    $\mathbf{C}(\mathcal{X})=\mathbf{C}_i$ if $g_i(\mathcal{X},\Lambda)=\max_j g_j(\mathcal{X},\Lambda)$

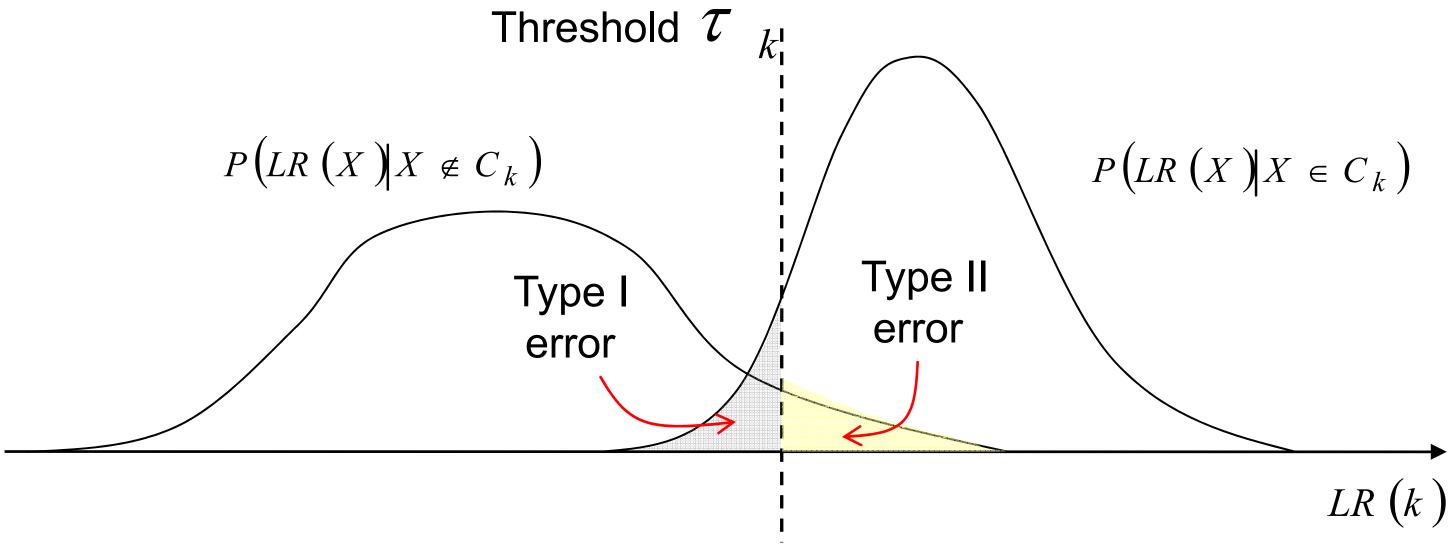
    $\Lambda=\{\lambda^{(i)}\}_{i=1......M}$ (M models, classes), $\mathcal{X}$: observations,

    $g_i(\mathcal{X},\Lambda)$: class conditioned likelihood function, for example,

    $$g_i(\mathcal{X},\Lambda)=\mathcal{P}(\mathcal{X}|\lambda^{(i)})$$
  - Traditional Training Criterion :

    find $\lambda^{(i)}$ such that $\mathcal{P}(\mathcal{X}|\lambda^{(i)})$ is maximum (Maximum Likelihood) if $\mathcal{X}\in\mathbf{C}_i$
    - This criterion does not always lead to minimum classification error, since **it doesn't consider the mutual relationship between different classes**
    - For example, it's possible that $\mathcal{P}(\mathcal{X}|\lambda^{(i)})$ is maximum but $\mathcal{X}\notin\mathbf{C}_i$

Minimum Classification Error (cont.)


Threshold τ_k
$P\left(LR(X) \middle| X \notin C_k\right)$
$P\left(LR(X) \middle| X \in C_k\right)$
Type I error
Type II error
$LR(k)$

Example showing histograms of the likelihood ratio
when the observation and

Type I error: False Rejection
Type II error: False Alarm/False Acceptance

Minimum Classification Error (cont.)

- Minimum Classification Error (MCE) (Cont.):
  - One form of the class misclassification measure :

$$d_i(X) = -g\left(X, \lambda^{(i)}\right) + \log\left[\frac{1}{M-1}\sum_{j \neq i} \exp\left(g\left(X, \lambda^{(i)}\right)\alpha\right)\right]^{\frac{1}{\alpha}} \qquad X \in C_i$$

$$d_i(X) \geq 0 \text{ implies a misclassification (error} = 1)$$

$$d_i(X) < 0 \text{ implies a correct classification (error} = 0)$$

  - A continuous loss function is defined as follows :

$$l_i(X, \Lambda) = l\left(d_i(X)\right) \quad X \in C_i$$

$$\textit{where the sigmoid function } l(d) = \frac{1}{1 + \exp(-\gamma d + \theta)}$$

  - Classifier performance measure :

$$L(\mathit{\Lambda}) = E_X\left[L(X, \mathit{\Lambda})\right] = \sum_X \sum_{i=1}^{M} l_i(X, \mathit{\Lambda})\delta(X \in C_i)$$

Minimum Classification Error (cont.)

- Using MCE in model training :
  - Find Λ such that

$$\hat{\Lambda} = \arg\min_{\Lambda} L(\Lambda) = \arg\min_{\Lambda} E_X \left[L(X, \Lambda) \right]$$

the above objective function in general cannot be minimized directly but the local minimum can be achieved using *gradient decent algorithm*

$$w_{t+1} = w_t - \varepsilon \frac{\partial L(\Lambda)}{\partial w}, w : an\ arbitrary\ \ parameter\ \ of\ \ \Lambda$$

- Using MCE in robust feature representation

$$\hat{f} = \arg\min_{f} E_X \left[L\left(f(X), \Lambda^{(f)} \right) \right]$$

f : a transform of the original feature X

$Note$: while feature presentation is changed, the model is also changed accordingly